

INFORMATIKA

Srednje splošno izobraževanje

Gimnazija

Gimnazija s slovenskim učnim jezikom na narodno mešanem območju v slovenski Istri

Gimnazija z italijanskim učnim jezikom na narodno mešanem območju v slovenski Istri

Dvojezična slovensko-madžarska gimnazija na narodno mešanem območju Prekmurja

Klasična gimnazija

Ekonomska gimnazija

Ekonomska gimnazija s slovenskim učnim jezikom na narodno mešanem območju v slovenski Istri

Tehniška gimnazija

Tehniška gimnazija s slovenskim učnim jezikom na narodno mešanem območju v slovenski Istri

Umetniška gimnazija

Umetniška gimnazija s slovenskim učnim jezikom na narodno mešanem območju v slovenski Istri



UČNI NAČRT Z DIDAKTIČNIMI PRIPOROČILI

IME PREDMETA: informatika

Izobraževalni program gimnazije, izobraževalni program gimnazije s slovenskim učnim jezikom na narodno mešanem območju v slovenski Istri, izobraževalni program gimnazije z italijanskim učnim jezikom na narodno mešanem območju v slovenski Istri, izobraževalni program dvojezične slovensko-madžarske gimnazije na narodno mešanem območju Prekmurja, izobraževalni program klasične gimnazije, izobraževalni program umetniške gimnazije - glasbena smer; modul A: glasbeni stavek, izobraževalni program umetniške gimnazije - glasbena smer; modul B: petje-instrument, izobraževalni program umetniške gimnazije - glasbena smer; modul C: jazz-zabavna glasba, izobraževalni program umetniške gimnazije - plesna smer; modul A: balet, izobraževalni program umetniške gimnazije - plesna smer; modul B: sodobni ples, izobraževalni program umetniške gimnazije - likovna smer, izobraževalni program umetniške gimnazije - smer gledališče in film, izobraževalni program umetniške gimnazije s slovenskim učnim jezikom na narodno mešanem območju v slovenski Istri - glasbena smer; modul A: glasbeni stavek, izobraževalni program umetniške gimnazije s slovenskim učnim jezikom na narodno mešanem območju v slovenski Istri - glasbena smer; modul B: petje-instrument, izobraževalni program umetniške gimnazije s slovenskim učnim jezikom na narodno mešanem območju v slovenski Istri - likovna smer:

obvezni predmet (70 ur)

maturitetni standard (280 ur)

Izobraževalni program ekonomske gimnazije, izobraževalni program ekonomske gimnazije s slovenskim učnim jezikom na narodno mešanem območju v slovenski Istri:

obvezni predmet (140 ur)

maturitetni standard (280 ur)

Izobraževalni program tehniške gimnazije, izobraževalni program tehniške gimnazije s slovenskim učnim jezikom na narodno mešanem območju v slovenski Istri:

obvezni predmet (105 ur)

maturitetni standard (280 ur)

PRIPRAVILA PREDMETNA KURIKULARNA KOMISIJA V SESTAVI:

dr. Andrej Brodnik, Univerza na Primorskem, Fakulteta za matematiko, naravoslovje in informacijsko tehnologijo; Maja Čelan, Srednja elektro-računalniška šola Maribor; Mirko Đukić, Škofijska gimnazija Antona Martina Slomška; mag. Radovan Krajnc, Zavod Republike Slovenije za šolstvo; mag. Matija Lokar, Univerza v Ljubljani, Fakulteta za matematiko in fiziko; Gregor Mede, Šolski center Novo mesto, Srednja elektro šola in tehniška gimnazija; Andrej Nemec, Osnovna šola Prežihovega Voranca Bistrica;

Boštjan Strnad, Osnovna šola Franceta Prešerna Maribor; dr. Boštjan Vlaovič, Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko

JEZIKOVNI PREGLED: Andraž Polončič Ruparčič

OBLIKOVANJE: neAGENCIJA, digitalne preobrazbe, Katja Pirc, s. p.

IZDALA: Ministrstvo za vzgojo in izobraževanje in Zavod RS za šolstvo

ZA MINISTRSTVO ZA VZGOJO IN IZOBRAŽEVANJE: dr. Vinko Logaj

ZA ZAVOD RS ZA ŠOLSTVO: Jasna Rojc

Ljubljana, 2025

SPLETNA IZDAJA

DOSTOPNO NA POVEZAVI:

https://eportal.mss.edus.si/msswww/datoteke/ucni_nacrti/2026/un-dp-informatika_gim-si-is-dv-kl-ek-ek-si-teh-teh-si-um-um-si.pdf

CIP

Kataložni zapis o publikaciji (CIP) pripravili v Narodni in univerzitetni knjižnici v Ljubljani

COBISS.SI-ID 247493891

ISBN 978-961-03-1037-2 (Zavod RS za šolstvo, PDF)

BESEDILO O SEJI SS

Strokovni svet RS za splošno izobraževanje je na svoji 244. seji, dne 22. 5. 2025, določil učni načrt informatika za izobraževalni program gimnazije, izobraževalni program gimnazije s slovenskim učnim jezikom na narodno mešanem območju v slovenski Istri, izobraževalni program gimnazije z italijanskim učnim jezikom na narodno mešanem območju v slovenski Istri, izobraževalni program dvojezične slovensko-madžarske gimnazije na narodno mešanem območju Prekmurja, izobraževalni program klasične gimnazije, izobraževalni program ekonomske gimnazije, izobraževalni program ekonomske gimnazije s slovenskim učnim jezikom na narodno mešanem območju v slovenski Istri, izobraževalni program tehniške gimnazije, izobraževalni program tehniške gimnazije s slovenskim učnim jezikom na narodno mešanem območju v slovenski Istri, izobraževalni program umetniške gimnazije in izobraževalni program umetniške gimnazije s slovenskim učnim jezikom na narodno mešanem območju v slovenski Istri.

Strokovni svet RS za splošno izobraževanje se je na svoji 244. seji, dne 22. 5. 2025, seznanil z didaktičnimi priporočili k učnemu načrtu informatika za izobraževalni program gimnazije, izobraževalni program gimnazije s slovenskim učnim jezikom na narodno mešanem območju v slovenski Istri, izobraževalni program gimnazije z italijanskim učnim jezikom na narodno mešanem območju v slovenski Istri, izobraževalni program dvojezične slovensko-madžarske gimnazije na narodno mešanem območju Prekmurja, izobraževalni program klasične gimnazije, izobraževalni program ekonomske gimnazije, izobraževalni program ekonomske gimnazije s

slovenskim učnim jezikom na narodno mešanem območju v slovenski Istri, izobraževalni program tehniške gimnazije, izobraževalni program tehniške gimnazije s slovenskim učnim jezikom na narodno mešanem območju v slovenski Istri, izobraževalni program umetniške gimnazije in izobraževalni program umetniške gimnazije s slovenskim učnim jezikom na narodno mešanem območju v slovenski Istri.



PRIZNANJE AVTORSTVA – NEKOMERCIALNO – DELJENJE POD ENAKIMI POGOJI

Prenova izobraževalnih programov s prenovo ključnih programskih dokumentov (kurikuluma za vrtce, učnih načrtov ter katalogov znanj)



REPUBLIKA SLOVENIJA
MINISTRSTVO ZA VZGOJO IN IZOBRAŽEVANJE



KAKO SE ZNAJTI?

TIPIČNE STRANI

V dokumentu je za dijakinje in dijake, učiteljice in učitelje ipd. uporabljena moška slovnična oblika, ki velja enakovredno za oba spola.

Tema sporoča vsebinski okvir ali okvir predmetnospecifičnih zmožnosti, povezanih s cilji oz. skupinami ciljev: teme so predmetno specifične in lahko vključujejo vsebinske sklope in dejavnosti ter razvijajo veščine, zmožnosti in koncepte, ki naj bi jih dijaki izgrajevali in razvijali v procesu učenja.

Vrsta teme:
Obvezno
Izbirno
Maturitetno

Opis teme zajema ključne poudarke oz. koncepte kot okvir skupinam ciljev.

Dodatna pojasnila za temo lahko vsebujejo razlago o zasnovi teme, usmeritve in/ali navodila, ki se nanašajo na vsebino, vrsto teme, členitev teme, poimenovanje, trajanje, dejavnosti, izvedbo aktivnosti, učne pripomočke ... oz. karkoli, kar je pomembno za to temo.

IME TEME

OBVEZNO

OPIS TEME

Em fugiaspiduci sitet ut etur, in conest dipsum eosa nonsequam volorum, ommolor aut desequi duciist estrunda volorest, sandeni maximin uscillab il magnam quiam quo con explacc ullaccum alisi quidell igendam, in repta dolupta volor sam, ut hilit laut faccumqui omnim ut doluptur, quam consequ atibus dolo dignate mporehent qui tem nonseque por aliquisimust que volestrum esed quibusdae milla conserf ernate dentiat.

Ma vellaut quatio rporrore voluptae cus et illent harum doluptatur, et quas molorporia doluptatis molum rae nos event, inullit incimaiozem fugiandantis et fugitisque maion ped quo odi beatur, sequi illisqui ipidel escim reprat.

DODATNA POJASNILA ZA TEMO

Ullatur sequo qui ut vellenderio. Re exceseq uiatur aturem eos millandio totaquis alibus iume voluptatia quis aut quamust iosant prestius, ut ommodigendis doluptaes dersped moluptae. Harum a volupta tiatur, ne veles quia voluptatia similic tatesequam, seque ventio. Hendis erum quissequo quist que con nulparum ut quuntemos et repro quod ma nimusa sae vellaut eum ulpari valent voluptu rempor a de omnimax impellu ptataecusa ducipsae aut evelles temolup tincil imporia simped et endes eum et molum que laborerat voluptiurem nempori bustio explabo rionsenet evenimin nis ex eum voluptam consed que volupid eliquibus, tetusda ntusam ellis ut archill uptiore rundiae nimusdam vel enit, optibea quiaspelitem lam alit re, omnis ab il mo dolupti commihi tionsequiam eicia dolo intur accusci musdaecatet faccatur? Sinctet latiam facescipsa

Ceria pa iumquam, exernam quaspis dit equasit, omnis quis aut apitassum coribusda invenih icatorum qui ut re perumquaepep quame moditatum ipient veligna tessint iaessum, optamus, testi sediasped moluptatem aperiam aut quodips usciate veribus.

Mi, asit volupta tuscil int aspel illit atur, eum idenist iorrum alit, nim re illitas ma nimi, iusclisita dolorrym volorepedio core, consequam quae. De nossiti officillate

Z oznako **SC** so označeni ključni cilji posameznih področij skupnih ciljev. Številčni zapis pri oznaki skupnih ciljev se navezuje na preglednico s skupnimi cilji.

27. 2. 2025 /// 15:45

IME SKUPINE CILJEV

CILJI

Z oznako **O** so označeni obvezni cilji.

Z ležečim tiskom in oznako **I** so označeni izbirni cilji.

- O:** Et plabo. Nam eum fuga. Et explat. Xerro et aut que nihicia doloribus, velisci tatur
I: *Que ommodic aboribust, tesequae nimaionseni arum audit minusda dolum, si venimil itatat. Estinve litatem ut adit quid qui.* **SC** (2.2.2.1)
I: *Quia voluptatur aut od minverum, quas ilis et mint, eature rehenit magnatur aut exceaquam reiunt et omnis nobit vero excepel in nimagmam quat officiam que plaborpore ne culpa eum dolore aut aspicias sanihil moditam re pelias ne veri reperferes solos dolorum, quidit ad qui dolorerun.*
O: Ibus nis auda vendio venis saperum atem quiant aut pre comniscia num at is quoditi ne estinve litatem venis saperum atem quiant. **SC** (1.3.4.2 | 2.1.1.1 | 2.1.2.1 | 2.2.1.1 | 2.2.1.2 | 3.1.3.1 | 5.1.2.1)

STANDARDI ZNANJA

Dijak

S krepko označenim besedilom so izpostavljeni **minimalni standardi znanja**.

Z ležečim besedilom so označeni izbirni standardi oz. standardi znanja, ki izhajajo iz izbirnih ciljev.

Dodatne pisave in oznake pri standardih so opredeljene v obvezujočih navodilih za učitelja (npr. podčrtani, M, PTI itd.)

- Obisquamet faccab incium repenam id quist plita nume con nonectem sit volentis etur sanda de velictorrum quasimus mo mi.
- Ilcillabo. Bor as plibusc ipidesequat latia cuptatu repeliam.
- Et ommo commolo ratatur saepedit essunt que parit, expelliquam rem simos con es est volorrum quae. Aximintios et faccae. Inciet, ommolo con commoluptas aut vidusci enient.**
- Ut fuga. Nemperum fuga.**
- Ma vellaut quiatio rporore voluptae cus et illent harum doluptatur, et quas
- Omnim ut doluptur, quam consequi.
- Obisquamet faccab incium repenam id quist plita nume con nonectem sit volentis etur sanda de velictorrum quasimus mo mi.*
- Ilcillabo. Bor as plibusc ipidesequat latia cuptatu repeliam.*
- Et ommo commolo ratatur saepedit essunt que parit, expelliquam rem simos con es est volorrum quae. Aximintios et faccae. Inciet, ommolo con commoluptas aut vidusci enient.
- Ut fuga. Nemperum fuga.*

TERMINI:

- | | | | |
|---|---|---|---------------------------------------|
| ☐ laborerat voluptiurem | ☐ veligna tessint | ☐ laborerat voluptiurem | ☐ veligna tessint |
| ☐ veligna tessint | ☐ laborerat voluptiurem | ☐ veligna tessint | |

Pojmi, ki jih dijak mora poznati oz. razumeti (in/ali znati smiselno uporabljati), lahko pa so navedeni tudi priporočeni pojmi. Nabor (strokovnih) terminov zaokrožuje in dopolnjuje posamezno temo, predstavlja pa informacijo učitelju pri načrtovanju vzgojno-izobraževalnega dela. Namenjen je navedbi terminologije, ki jo na področju določene teme v določenem obdobju izobraževalne vertikale usvojijo dijaki.

VIRI IN LITERATURA PO POGLAVJIH

Viri in literatura,
ki so vključeni v
posamezno temo.

NASLOV TEME

- » Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi eget ante eget ante hendrerit convallis vitae et purus. Donec euismod dolor sed neque condimentum, at sodales enim interdum. <https://lorem.ipsum.com/mubis/link/n0n2c3c>

NASLOV SKUPINE CILJEV

- » Etiam, M. (2019). Eget laoreet ipsum. Nulla facilisi.
- » Parum, V. (2025). Sequi blaboreet quosam. Seque cus.
- » Morbi P. (2007). Sed ac arcu id velit facilisis aliquet. Nec tincidunt.
- » Eget O. (2010). Vestibulum vitae massa eget orci condimentum fringilla a ac turpis. Nulla et tempus nunc, vitae scelerisque tellus. Sed in tempus mi, a aliquet quam. Integer ut euismod eros, vel pretium mi. Maecenas sollicitudin.
- » Nunc N. (2023). Morbi suscipit ante quis viverra sollicitudin. Aenean ultricies.
- » Nunc N., Ante A., Sem S. (2015). Nullam congue eleifend magna in venenatis. Ut consectetur quis magna vitae sodales. Morbi placerat: <https://lorem.ipsum.com/mubis/link/n0n2c3c>
- » Sed A. (2024). Ut aliquet aliquam urna eget laoreet. Fusce hendrerit dolor id mauris convallis, at accumsan lectus fermentum. Nunc bibendum quam et nibh elementum consectetur.

NASLOV SKUPINE CILJEV

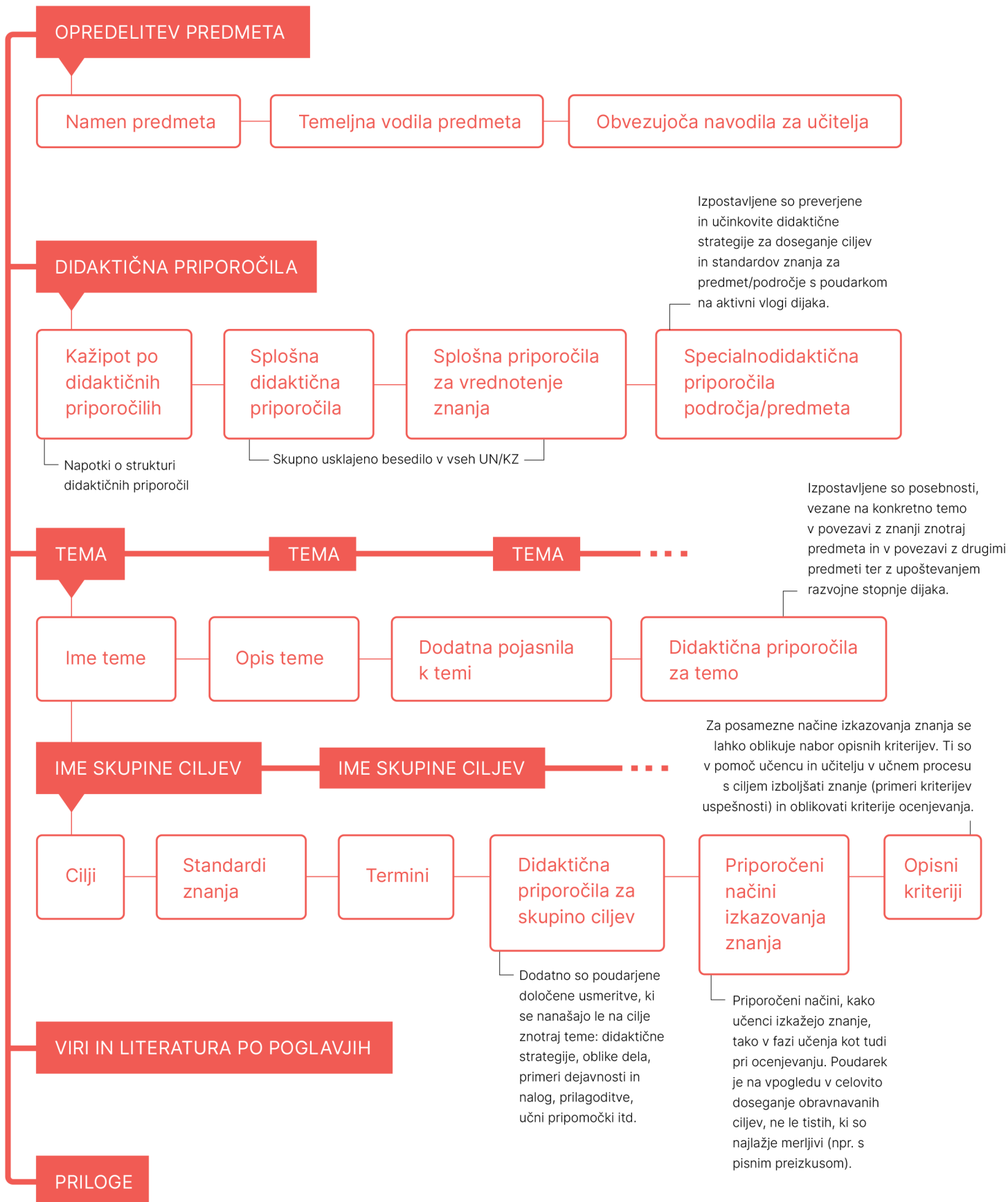
- » Vestibulum vitae massa eget orci condimentum fringilla, <https://lorem.ipsum.com/mubis/link/n0n2c3c>
- » Nunc F.: Vivamus quis tortor et ipsum tempor laoreet id quis nunc. Phasellus quis sagittis ligula.
- » Redaktion, 2018. Morbi placerat: <https://lorem.ipsum.com/mubis/link/n0n2c3c>

Viri in literatura, ki so vključeni
v posamezno skupino ciljev.

Hiperpovezava
do zunanjega vira

KAKO SE ZNAJTI?

STRUKTURA UČNEGA NAČRTA Z DIDAKTIČNIMI PRIPOROČILI



KAZALO

OPREDELITEV PREDMETA.....	10		
Namen predmeta.....	10		
Temeljna vodila predmeta	10		
Obvezujoča navodila za učitelje	11		
DIDAKTIČNA PRIPOROČILA	12		
Kažipot po didaktičnih priporočilih	12		
Splošna didaktična priporočila	12		
Splošna priporočila za vrednotenje znanja ...	14		
Specialnodidaktična priporočila področja/predmeta	15		
TEME, CILJI, STANDARDI ZNANJA.....	21		
UČINKI RAČUNALNIŠTVA IN INFORMATIKE	22		
Učinki računalništva in informatike 70	22		
Učinki računalništva in informatike 210	25		
Vpliv na naravo in vsebino poklicev 70.....	27		
RAČUNALNIŠKI SISTEMI.....	30		
Von Neumannov model 70.....	30		
Von Neumannov model 210.....	33		
Sistemska programska oprema 70.....	35		
Sistemska programska oprema 210.....	36		
Digitalizacija podatkov 70	37		
Digitalizacija podatkov 210	39		
Prenos in pravilnost zapisa podatkov 210.....	41		
PODATKI IN ANALIZA	42		
Zbiranje in shranjevanje 70	44		
Zbiranje in shranjevanje 210	46		
Prikazovanje in preoblikovanje 70.....	49		
Prikazovanje in preoblikovanje 210.....	53		
Sklepanje in modeliranje 70.....	57		
Sklepanje in modeliranje 210.....	58		
ALGORITMI IN PROGRAMIRANJE.....	61		
Algoritmi 70	75		
Algoritmi 210	77		
Spremenljivke 70	78		
Spremenljivke 210	80		
Kontrolne strukture 70.....	81		
		Kontrolne strukture 210.....	83
		Modularnost 70	84
		Modularnost 210	86
		Razvoj programov 70	88
		Razvoj programov 210	91
		Izbirni cilji 210.....	93
		RAČUNALNIŠKA OMREŽJA	95
		Namen in zgradba omrežij 70.....	96
		Namen in zgradba omrežij 210	98
		Prenos sporočil 70.....	100
		Prenos sporočil 210.....	101
		Varnost 70	103
		Varnost 210.....	105
		PRILOGE	107

OPREDELITEV PREDMETA

NAMEN PREDMETA

Informatika je predmet, namenjen pridobivanju temeljnih znanj o osnovnih konceptih, na katerih temelji digitalni svet, in razvijanju računalniškega mišljenja, ki predstavlja temelj interakcije z njim. S pridobljenim znanjem lahko dijaki razvijajo svoje potencialne, uresničujejo lastne ideje in se učijo spoprijemati z izzivi digitalne prihodnosti. Pri predmetu informatika dijaki dobijo podlago za aktivno, odgovorno in varno delovanje v sodobni družbi.

TEMELJNA VODILA PREDMETA

Dijaki v avtentičnih situacijah prepoznajo probleme in presojujejo, ali jih je mogoče učinkovito rešiti z računalnikom. Urijo se v iskanju vzorcev v problemih, v razumevanju struktur problemov in iskanju rešitev. Rešitve razvijajo s pomočjo abstrakcije in izdelave modelov. Z njimi zmanjšujemo kompleksnost predstavitev problemov ter jih naredijo obvladljivejše. Za izdelane modele razvijajo algoritme, ki jih nato lahko implementirajo v programskem jeziku. Skupaj z iterativnim procesom razvoja in testiranja dijaki postopoma izboljšujejo svoje modele, algoritme ali programe, dokler ne dosežejo želenih rezultatov ali rešitve problemov. Tak pristop dijakom omogoča praktično razumevanje informacijskih procesov, delovanja računalniških sistemov in reševanja problemov z računalnikom.

Dijaki razvijajo zmožnost sodelovanja v skupinah. Urijo se v oblikovanju skupinskih norm, dogovarjanju o načinih sodelovanja in razdelitvi nalog, tako da se prilagodijo specifikam skupine in problemom, ki jih rešujejo.

Pri informatiki morajo imeti dijaki več celovitih izkušenj reševanja problemov z digitalno tehnologijo, pri čemer lahko določena znanja pridobivajo tudi z aktivnostmi brez zaslonov (t. i. računalništvo brez računalnika). Celovitost zajema celotno pot od določitve avtentičnega problema do končne rešitve.

OBVEZUJOČA NAVODILA ZA UČITELJE

Učni načrt za predmet *informatika* v programih: ***gimnazija, klasična gimnazija in umetniška gimnazija*** je zasnovan tako, da loči cilje, ki naj bi jih dosegli učenci v obveznih 70 urah, od ciljev, ki naj bi jih dosegli učenci, ki so predmet informatika izbrali na maturi.

- » Skupine ciljev, označene s številko 70, morajo doseči vsi dijaki (običajno v prvem letniku).
- » Skupine ciljev, označene s številko 210, so namenjene tistim dijakom, ki so informatiko izbrali kot maturitetni predmet.

V programu ***ekonomska gimnazija*** je obveznemu delu pouka *informatike* namenjenih 140 ur. Učitelj mora v tem delu realizirati vse cilje, označene z oznako 70. V preostalih 70 urah (do skupno 140 ur) predlagamo, da učitelj izbere dodatne cilje iz obsežnejšega nabora, označenega z oznako 210. Priporočamo, da učitelj te cilje izbira iz naslednjih tem:

- » Učinki računalništva in informatike,
- » Podatki in analiza,
- » Algoritmi in programiranje.

V programu ***tehniška gimnazija*** je obveznemu delu pouka *informatike* namenjenih 105 ur. V sklopu teh ur mora učitelj realizirati vse cilje, označene z oznako 70. V preostalih 35 urah (do skupno 105 ur) naj učitelj izbere dodatne cilje iz obsežnejšega nabora 210. Pri tem priporočamo, da te dodatne cilje izbira iz teme Podatki in analiza.



DIDAKTIČNA PRIPOROČILA

KAŽIPOT PO DIDAKTIČNIH PRIPOROČILIH

Razdelke *Kažipot po didaktičnih priporočilih*, *Splošna didaktična priporočila* in *Splošna priporočila za vrednotenje znanja* je pripravil Zavod RS za šolstvo.

Didaktična priporočila prinašajo učiteljem napotke za uresničevanje učnega načrta predmeta v pedagoškem procesu. Zastavljena so večplastno, na več ravneh (od splošnega h konkretnemu), ki se medsebojno prepletajo in dopolnjujejo.

- » Razdelka **Splošna didaktična priporočila** in **Splošna priporočila za vrednotenje znanja** vključujeta krovne usmeritve za načrtovanje, poučevanje in vrednotenje znanja, ki veljajo za vse predmete po celotni izobraževalni vertikali. Besedilo v teh dveh razdelkih je nastalo na podlagi *Usmeritev za pripravo didaktičnih priporočil k učnim načrtom za osnovne šole in srednje šole* (<https://aplikacijaun.zrss.si/api/link/3ladrdr>) ter *Izhodišč za prenovo učnih načrtov v osnovni šoli in gimnaziji* (<https://aplikacijaun.zrss.si/api/link/plw0909>) in je v vseh učnih načrtih enako.
- » Razdelek **Specialnodidaktična priporočila področja/predmeta** vključuje tista didaktična priporočila, ki se navezujejo na področje/predmet kot celoto. Zajeti so didaktični pristopi in strategije, ki so posebej priporočeni in značilni za predmet glede na njegovo naravo in specifične.

Učni načrt posameznega predmeta je členjen na *teme*, vsaka tema pa se lahko nadalje členi na *skupine ciljev*.

- » Razdelka **Didaktična priporočila za temo** in **Didaktična priporočila za skupino ciljev** vključujeta konkretne in specifične napotke, ki se nanašajo na poučevanje določene teme oz. skupine ciljev znotraj teme. Na tem mestu so izpostavljene preverjene in učinkovite didaktične strategije za poučevanje posamezne teme ob upoštevanju značilnosti in vidikov znanja, starosti dijakov, predznanja, povezanosti znanja z drugimi predmeti/področji ipd. Na tej ravni so usmeritve lahko konkretizirane tudi s primeri izpeljave oz. učnimi scenariji.

Didaktična priporočila na ravni skupine ciljev zaokrožujeta razdelka **Priporočeni načini izkazovanja znanja** in **Opisni kriteriji**, ki vključujeta napotke za vrednotenje znanja (spremljanje, preverjanje, ocenjevanje) znotraj posamezne teme oz. skupine ciljev.

SPLOŠNA DIDAKTIČNA PRIPOROČILA

Učitelj si za uresničitev ciljev učnega načrta, kakovostno učenje ter optimalni psihofizični razvoj dijakov prizadeva zagotoviti varno in spodbudno učno okolje. V ta namen pri poučevanju uporablja raznolike

didaktične strategije, ki vključujejo učne oblike, metode, tehnike, učna sredstva in gradiva, s katerimi dijakom omogoča aktivno sodelovanje pri pouku, pa tudi samostojno učenje. Izbira jih premišljeno, glede na namen in naravo učnih ciljev ter glede na učne in druge, za učenje pomembne značilnosti posameznega dijaka, učne skupine ali oddelka.

Varno in spodbudno učno okolje učitelj zagotavlja tako, da:

- » spodbuja medsebojno sprejemanje, sodelovanje, čustveno in socialno podporo;
- » neguje vedoželjnost, spodbuja interes in motivacijo za učenje, podpira razvoj različnih talentov in potencialov;
- » dijake aktivno vključuje v načrtovanje učenja;
- » kakovostno poučuje in organizira samostojno učenje (individualno, v parih, skupinsko) ob različni stopnji vodenja in spodbujanja;
- » dijakom omogoča medsebojno izmenjavo znanja in izkušenj, podporo in sodelovanje;
- » prepozna in pri poučevanju upošteva predznanje, skupne in individualne učne, socialne, čustvene, (med)kulturne, telesne in druge potrebe dijakov;
- » dijakom postavlja ustrezno zahtevne učne izzive in si prizadeva za njihov napredek;
- » pri dijakih stalno preverja razumevanje, spodbuja ozaveščanje in usmerjanje procesa lastnega učenja;
- » proces poučevanja prilagaja ugotovitvam sprotne spremljanja in preverjanja dosežkov dijakov;
- » omogoča povezovanje ter nadgrajevanje znanja znotraj predmeta, med predmeti in predmetnimi področji;
- » poučuje in organizira samostojno učenje v različnih učnih okoljih (tudi virtualnih, zunaj učilnic), ob uporabi avtentičnih učnih virov in reševanju relevantnih življenjskih problemov in situacij;
- » ob doseganju predmetnih uresničuje tudi skupne cilje različnih področij (jezik, državljanstvo, kultura in umetnost; trajnostni razvoj; zdravje in dobrobit; digitalna kompetentnost; podjetnost).

Učitelj pri uresničevanju ciljev in standardov znanja učnega načrta dijakom omogoči prepoznavanje in razumevanje:

- » smisla oz. namena učenja (kaj se bodo učili in čemu);
- » uspešnosti lastnega učenja oz. napredka (kako in na temelju česa bodo vedeli, da so pri učenju uspešni in so dosegli cilj);
- » pomena različnih dokazov o učenju in znanju;
- » vloge povratne informacije za stalno izboljševanje ter krepitev občutka »zmorem«;
- » pomena medvrstniškega učenja in vrstniške povratne informacije.

Za doseganje celostnega in poglobljenega znanja učitelj načrtuje raznolike predmetne ali medpredmetne učne izzive, ki spodbujajo dijake k aktivnemu raziskovanju, preizkušanju, primerjanju, analiziranju, argumentiranju,

reševanju avtentičnih problemov, izmenjavi izkušenj in povratnih informacij. Ob tem nadgrajujejo znanje ter razvijajo ustvarjalnost, inovativnost, kritično mišljenje in druge prečne veščine. Zato učitelj, kadar je mogoče, izvaja projektni, problemski, raziskovalni, eksperimentalni, izkustveni ali praktični pouk in uporablja temu primerne učne metode, pripomočke, gradiva in digitalno tehnologijo.

Učitelj upošteva raznolike zmožnosti in potrebe dijakov v okviru notranje diferenciacije in individualizacije pouka ter personalizacije učenja s prilagoditvami, ki obsegajo:

- » učno okolje z izbiro ustreznih didaktičnih strategij, učnih dejavnosti in oblik;
- » obsežnost, zahtevnost in kompleksnost učnih ciljev;
- » raznovrstnost in tempo učenja;
- » načine izkazovanja znanja, pričakovane rezultate ali dosežke.

Učitelj smiselno upošteva načelo diferenciacije in individualizacije tudi pri načrtovanju domačega dela dijakov, ki naj bo osmišljeno in raznoliko, namenjeno utrjevanju znanja in pripravi na nadaljnje učenje.

Individualizacija pouka in personalizirano učenje sta pomembna za razvijanje talentov in potencialov nadarjenih dijakov. Še posebej pa sta pomembna za razvoj, uspešno učenje ter enakovredno in aktivno vključenost dijakov s posebnimi vzgojno-izobraževalnimi potrebami, z učnimi težavami, dvojno izjemnih, priseljencev ter dijakov iz manj spodbudnega družinskega okolja. Z individualiziranimi pristopi preko inkluzivne poučevalne prakse učitelj odkriva in zmanjšuje ovire, ki dijakom iz teh skupin onemogočajo optimalno učenje, razvoj in izkazovanje znanja, ter uresničuje v individualiziranih programih in v drugih individualiziranih načrtih načrtovane prilagoditve vzgojno-izobraževalnega procesa za dijake iz specifičnih skupin.

SPLOŠNA PRIPOROČILA ZA VREDNOTENJE ZNANJA

Vrednotenje znanja razumemo kot ugotavljanje znanja dijakov skozi celoten učni proces, tako pri spremljanju in preverjanju (ugotavljanje predznanja in znanja dijaka na vseh stopnjah učenja) kot tudi pri ocenjevanju znanja.

V prvi fazi učitelj kontinuirano spremlja in podpira učenje, preverja znanje vsakega dijaka, mu nudi kakovostne povratne informacije in ob tem ustrezno prilagaja lastno poučevanje. Pred začetkom učnega procesa učitelj najprej aktivira in ugotavlja dijakovo predznanje in ugotovitve uporabi pri načrtovanju pouka. Med učnim procesom sproti preverja doseganje ciljev pouka in standardov znanja ter spremlja in ugotavlja napredek dijaka. V tej fazi učitelj znanja ne ocenjuje, pač pa na osnovi ugotovitev sproti prilagaja in izvaja dejavnosti v podporo in spodbudo učenju (npr. dodatne dejavnosti za utrjevanje znanja, prilagoditve načrtovanih dejavnosti in nalog glede na zmožnosti in potrebe posameznih dijakov ali skupine).

Učitelj pripomore k večji kakovosti pouka in učenja, tako da:

- » sistematično, kontinuirano in načrtno pridobiva informacije o tem, kako dijak dosega učne cilje in usvaja standarde znanja;
- » ugotavlja in spodbuja razvoj raznolikega znanja – ne le vsebinskega, temveč tudi procesnega (tj. spretnosti in veščin), spremlja in spodbuja pa tudi razvijanje odnosnega znanja;

- » spodbuja dijaka, da dosega cilje na različnih taksonomskih ravneh oz. izkazuje znanje na različnih ravneh zahtevnosti;
- » spodbuja uporabo znanja za reševanje problemov, sklepanje, analiziranje, vrednotenje, argumentiranje itn.;
- » je naravnani na ugotavljanje napredka in dosežkov, pri čemer razume, da so pomanjkljivosti in napake zlasti priložnosti za nadaljnje učenje;
- » ugotavlja in analizira dijakovo razumevanje ter odpravlja vzroke za nerazumevanje in napačne predstave;
- » dijaka spodbuja in ga vključuje v premisleke o namenih učenja in kriterijih uspešnosti, po katerih vrednoti lastno učno uspešnost (samovrednotenje) in uspešnost vrstnikov (vrstniško vrednotenje);
- » dijaku sproti podaja kakovostne povratne informacije, ki vključujejo usmeritve za nadaljnje učenje.

Ko so dejavnosti prve faze (spremljanje in preverjanje znanja) ustrezno izpeljane, sledi druga faza, ocenjevanje znanja. Pri tem učitelj dijaku omogoči, da lahko v čim večji meri izkaže usvojeno znanje. To doseže tako, da ocenjuje znanje na različne načine, ki jih je dijak spoznal v procesu učenja. Pri tem upošteva potrebe dijaka, ki za uspešno učenje in izkazovanje znanja potrebuje prilagoditve.

Učitelj lahko ocenjuje samo znanje, ki je v učnem načrtu določeno s standardi znanja. Predmet ocenjevanja znanja niso vsi učni cilji, saj vsak cilj nima z njim povezanega specifičnega standarda znanja. Učitelj ne ocenjuje stališč, vrednot, navad, socialnih in čustvenih veščin ipd., čeprav so te zajete v ciljih učnega načrta in jih učitelj pri dijaku sistematično spodbuja, razvija in v okviru prve faze tudi spremlja.

Na podlagi standardov znanja in kriterijev uspešnosti učitelj, tudi v sodelovanju z drugimi učitelji, pripravi kriterije ocenjevanja in opisnike ter jih na ustrezen način predstavi dijaku. Če dijak v procesu učenja razume in uporablja kriterije uspešnosti, bo lažje razumel kriterije ocenjevanja. Ugotovitve o doseganju standardov znanja, ki temeljijo na kriterijih ocenjevanja in opisnikih, se izrazijo v obliki ocene.

Učitelj z raznolikimi načini ocenjevanja omogoči izkazovanje raznolikega znanja (védenje, spretnosti, veščine) na različnih ravneh. Zato poleg pisnih preizkusov znanja in ustnih odgovorov ocenjuje izdelke (pisne, likovne, tehnične, praktične in druge za predmet specifične) in izvedbo dejavnosti (govorne, gibalne, umetniške, eksperimentalne, praktične, multimedijske, demonstracije, nastope in druge za predmet specifične), s katerimi dijak izkaže svoje znanje.

SPECIALNODIDAKTIČNA PRIPOROČILA PODROČJA/PREDMETA

Učinkovito poučevanje informatike zahteva več kot le poučevanje temeljnih znanj računalništva in informatike. Od učiteljev zahteva premišljene didaktične pristope, ki dijakom omogočajo poglobljeno razumevanje računalniških konceptov, razvoj analitičnega mišljenja in ustvarjalnost pri reševanju problemov. V nadaljevanju bomo navedli 12 pedagoških pristopov in smernic, kako strukturirati pouk, prilagoditi metode poučevanja in spodbujati aktivno sodelovanje dijakov. Ti pristopi temeljijo na raziskavah in dobrih praksah, ki dokazano

izboljšujejo razumevanje računalniških konceptov ter krepijo motivacijo dijakov. S tem učitelji ustvarjajo spodbudno učno okolje, v katerem dijaki ne pridobijo le veščin in znanja, temveč razvijejo tudi računalniško mišljenje in samozavest pri delu z digitalnimi tehnologijami (vir: Raspberry Pi foundation, The Big Book of Computing Pedagogy, 2021 (<https://aplikacijaun.zrss.si/api/link/sldcd5c>)).

1. **Začnite s koncepti** – Preden se dijaki lotijo pisanja kode, naj razumejo osnovne računalniške koncepte. Pomembno je, da vidijo širšo sliko, na primer kaj so algoritmi, podatkovne strukture ali abstrakcija, preden se poglobijo v podrobnosti sintakse določenega programskega jezika.

Strukturirajte pouk – Pouk mora imeti jasno strukturo, ki dijakom omogoča postopno grajenje znanja. Učno pot načrtujte tako, da se novi koncepti povezujejo s predhodnim znanjem in da dijaki vidijo logično zaporedje vsebin.

2. **Računalniške koncepte ponazorite** – Abstraktne ideje so za dijake pogosto težavne, zato jih predstavite na konkreten način. Uporabljajte analogije, vizualizacije, praktične primere ali fizične predmete, ki pomagajo dijakom razumeti sicer zahtevne pojme.
3. **Uporabite dejavnosti računalništva brez računalnika, razčlenitev in ponovno sestavljanje** – Pred konkretnim delom z računalnikom lahko dijaki spoznavajo koncepte prek dejavnosti brez računalnika (npr. algoritmi na papirju ali v prostoru). Nato razčlenite probleme na manjše dele, jih analizirajte in na koncu povežite v smiselno celoto.
4. **Spodbujajte sodelovanje** – Učenje v skupinah omogoča izmenjavo idej, razvoj komunikacijskih veščin in boljše razumevanje. Pari ali skupine naj skupaj rešujejo naloge, analizirajo kodo in si medsebojno razlagajo pojme.
5. **Najprej berite in raziskujte kodo** – Pred prvim pisanjem kode naj dijaki analizirajo že obstoječo kodo. To jim pomaga razumeti strukturo programov, prepoznati vzorce in se učiti iz primerov. Lahko jim ponudite kodo z napakami ali nepopolne primere, ki jih morajo dopolniti.
6. **Spodbujajte razumevanje programov** – Učenje programiranja ni samo pisanje kode, ampak tudi njeno razumevanje. Uporabite strategije, kot so urejanje zaporedja ukazov (Parsonsovi problemi), razlaga kode s komentarji ali popravljanje obstoječih programov.
7. **Modelirajte vse** – Učitelj naj pokaže, kako razmišlja pri reševanju problemov. Na glas govorite, kako načrtujete rešitev, odpravljate napake ali analizirate rezultate. Tako dijaki spoznajo, kako razmišljajo eksperti.

8. **Odpravite napačne predstave** – Dijaki imajo pogosto napačno predznanje ali predstave, ki niso pravilne. Pomembno je, da jih učitelj prepozna in jih skozi razpravo ter praktične primere pomaga razjasniti.
9. **Omogočite projektno delo** – Projekti dijakom omogočajo, da uporabijo naučene koncepte v praksi. Spodbujajte jih, da razvijejo lastne aplikacije, igre ali interaktivne zgodbe, saj to krepi motivacijo in utrjuje znanje.
10. **Omogočite praktične izkušnje** – Aktivno učenje je ključno. Naj dijaki eksperimentirajo, popravljajo napake, preizkušajo različne pristope in raziskujejo rešitve. S tem pridobijo samozavest in razumejo delovanje digitalne tehnologije v praksi.
11. **Vpeljite raznolikost metod** – Uporabljajte različne učne metode, kot so igrifikacija, vizualizacije, razprave, uganke in praktične naloge. Različni pristopi pomagajo doseči različne učence in naredijo pouk zanimivejši.

Ti pristopi omogočajo predstavitev področja računalništva in informatike na način, ki je logičen, zanimiv in prilagojen dijakovim potrebam.

Učitelj presodi število ur, ki jih bo namenil posamezni temi. Dijaki morajo spoznati vse teme. Spodnji časovni okvir je zgolj predlog, učitelj pa je avtonomen, da se glede na predznanje dijakov in njihove potrebe odloči glede okvirnega števila ur za posamezno skupino ciljev.

Tema	70	210
Učinki računalništva	4	10
Računalniški sistemi	16	40
Podatki in analiza	8	40
Algoritmi in programiranje	28	60
Računalniška omrežja	6	30
Nerazporejene ure, ocenjevanje	8	30

Razvijanje področja skupnih ciljev:

Jezikovna kompetenca

Pri razvoju strokovnega jezika v okviru predmeta informatika je ključno, da dijaki spoznavajo slovenske strokovne izraze s področja računalništva in informatike. Če bodo dijaki strokovne izraze **SC** (1.1.2.2) večkrat slišali v različnih kontekstih, jih bodo lažje spoznali, razumeli in uporabljali?

Tudi pri ustvarjanju digitalnih izdelkov spodbujajte dijake k ustreznemu govornemu in pisnemu izražanju. S tem boste poleg jezikovne kompetence razvijali tudi globlje razumevanje vsebin predmeta.

Znanje jezika omogoča ne le uspešno komunikacijo, temveč tudi razvoj kritičnega mišljenja, saj dijaki ob branju strokovnih besedil razvijajo sposobnost analitičnega in natančnega izražanja svojih misli. Poleg tega bo dijak

spoznal, da je učenje vsebine posameznega predmeta hkrati tudi spoznavanje njegove strokovne terminologije. (SC (1.1.2.1))

Poudarek na strokovnih izrazih v slovenščini bo prispeval k bogatenju dijakovega jezikovnega repertoarja ter k uspešnemu vključevanju v večjezično družbo. Preko ustvarjanja digitalnih besedil in branja strokovnih člankov bodo dijaki razvijali svojo sposobnost tvorjenja in razumevanja strokovnih besedil, kar bo prispevalo k njihovi celoviti pismenosti. S tem bo poskrbljeno za celostni razvoj kompetenc, ki so ključne ne le za uspeh pri predmetu, temveč tudi za njihovo aktivno in odgovorno vlogo v sodobni družbi.

Duševna dobrobit in prožen način mišljenja

Za razvijanje kompetence, ki vključuje osebno prožnost, razvojno miselno naravnost, radovednost, optimizem in ustvarjalnost (SC (3.1.4.1)), mora učitelj ustvariti okolje, ki spodbuja celostni razvoj dijakov ter krepi njihovo duševno, telesno in socialno dobrobit. To se lahko dosega skozi različne učne vsebine, spodbudne učne strategije in varno ter podpirajoče učno okolje.

Učitelj naj z uporabo pristopov, ki temeljijo na aktivnem učenju, dijake vodi skozi procese samozavedanja in samorefleksije, kjer dijaki prepoznavajo svoje prednosti in slabosti ter postavljajo osebne in učne cilje. Pomembno je spodbujati razvojno miselno naravnost, kjer se dijaki učijo dojemati napake kot priložnost za rast, s tem pa razvijajo osebno prožnost (pri temi Algoritmi in programiranje pokažemo, da je napaka del poti na poti k rešitvi). Preko nalog in projektov, ki vključujejo elemente raziskovanja in eksperimentiranja, učitelj krepi radovednost in ustvarjalnost dijakov ter jih spodbuja k iskanju inovativnih rešitev.

Ustvarjanje varnega in spodbudnega učnega okolja je ključno za razvoj teh kompetenc. To vključuje oblikovanje odprtega prostora za diskusijo, kjer lahko dijaki izražajo svoja mnenja brez strahu pred kritiko. Učitelj naj podpira tudi optimizem in pozitivno naravnost, zlasti z motiviranjem dijakov k vztrajnosti, tudi ko se soočajo s težavami.

Učitelji so ključni pri oblikovanju šolskega okolja, ki dolgoročno prispeva k večji kakovosti življenja učencev. To se bo odražalo v njihovi sposobnosti konstruktivnega spoprijemanja z izzivi in v razvoju celovite osebnosti.

Telesna dobrobit in gibanje ter sedenje

Za ozaveščanje škodljivosti dolgotrajnega sedenja in spodbujanje zdravih navad (SC (3.2.1.4)) je ključno, da učitelj vodi z zgledom ter organizira pouk na način, ki dijakom omogoča prekinitve sedenja in kratko sprostitev telesa ter misli. To lahko učitelj doseže na različne načine:

- » **Vključevanje aktivnih odmorov:** Pouk se prekine z aktivnimi odmori, ki so kratki (2–3 minute), a dovolj intenzivni, da spodbujajo prekrvavitev in razgibajo telo. Vaje, kot so raztezne, dihalne vaje, sproščanje oči ali sprehod po učilnici, naj bodo prilagojene dijakom in ne zahtevajo posebne opreme. Aktivni odmori naj postanejo del redne rutine.
- » **Drugačne oblike dela:** Učitelj organizira delo v skupinah ali projektno delo, kjer dijaki ne sedijo ves čas. Naloge lahko vključujejo menjavo mest v razredu ali izvajanje aktivnosti stoje.

- » **Gibalno dejavno učenje:** Pouk poteka v kombinaciji z gibanjem, kar prispeva k aktivnejšemu pristopu k učenju.
- » **Pouk na prostem:** Učitelj občasno organizira dejavnosti zunaj učilnice, kar omogoča gibanje na svežem zraku.

Vključitev takšnih pristopov izboljša duševno, telesno in socialno dobrobit dijakov ter spodbuja razvoj zdravih življenjskih navad, ki jih bodo spremljale tudi zunaj šolskega okolja.

Razvijanje podjetnostne kompetence

Podjetnostna kompetenca se razvija celostno, skozi reševanje avtentičnih izzivov in povezovanje različnih znanj ter veščin. Pri pouku informatike lahko dijaki razvijajo podjetnost pri temah Algoritmi in programiranje ter Podatki in analiza. Pri tem učitelj pripravi dejavnosti, ki vključujejo prepoznavanje problemov in iskanje rešitev, ki ustvarjajo vrednost za druge. Dijaki naj sodelujejo pri načrtovanju, razvijanju in implementaciji rešitev, pri čemer se osredotočajo na ustvarjalnost, sodelovanje, kritično razmišljanje in odgovornost.

Praktične dejavnosti za spodbujanje podjetnosti:

12. **Identifikacija problemov:** Dijaki prepoznajo izzive  (5.1.1.1) iz vsakdanjega življenja, šolske skupnosti ali širšega okolja.
13. **Razvoj idej in rešitev:** Skupine dijakov pripravijo več mogočih rešitev,  (5.1.2.2) pri čemer uporabljajo različna orodja in tehnike, kot so miselni vzorci, analiza SWOT ali prototipiranje aplikacij.
14. **Uresničevanje zamisli:** Dijaki svoje zamisli prenesajo v rešitve, na primer s programiranjem ali zbiranjem, obdelavo, prikazom ter interpretacijo podatkov.  (5.2.3.1)
15. **Povratne informacije in izboljšave:** Dijaki predstavijo svoje rešitve in prejmejo povratne informacije od sošolcev, učiteljev ali zunanjih strokovnjakov.  (5.3.5.2) Nato rešitve prilagodijo glede na pridobljene povratne informacije.

Vloga učitelja

Učitelj ima ključno vlogo kot mentor, ki spodbuja raziskovanje, postavljanje vprašanj in sodelovanje. Aktivnosti naj bodo strukturirane tako, da omogočajo postopno razvijanje posameznih kompetenc (npr. ustvarjalnosti, sodelovanja, odločanja) kot gradnikov podjetnosti. Učitelj naj dijake vodi pri oblikovanju ciljev, načrtovanju dela in spremljanju njihovega napredka.

Z vključevanjem takšnih dejavnosti se podjetnostna kompetenca razvija naravno in smiselno, saj dijaki znanja ne le pridobivajo, temveč jih aktivno uporabljajo za reševanje konkretnih izzivov.

Razvijanje digitalne kompetentnosti

Razvoj digitalne kompetentnosti je naloga vseh predmetov in učiteljev, saj digitalne tehnologije prežemajo vse vidike učenja in dela. Vsak učitelj prispeva k razvoju temeljne digitalne kompetentnosti pri dijakih, ne glede na predmet, ki ga poučuje.

Posebnost predmeta informatika pa je, da omogoča poglobljeno razumevanje digitalnih tehnologij, njihovega delovanja in vpliva. Prav zato lahko pri informatiki nekatere digitalne kompetence razvijamo na višjih nivojih kot pri drugih predmetih. Dijaki ne bodo le uporabniki digitalnih orodij, temveč bodo razumeli tudi njihove zakonitosti, delovanje in omejitve, kar jim bo omogočalo kritično in ustvarjalno rabo digitalne tehnologije.

V učnem načrtu informatike zato namenjamo posebno pozornost naslednjim digitalnim kompetencam iz okvirja DigComp 2.2 (s klikom na ime kompetence se odpre njena razlaga):

- » 1.2 Vrednotenje podatkov (<https://aplikacijaun.zrss.si/api/link/f2l5v89>), informacij in digitalnih vsebin,
- » 1.3 Upravljanje podatkov (<https://aplikacijaun.zrss.si/api/link/0j1aogo>), informacij in digitalnih vsebin,
- » 3.3 Avtorske pravice (<https://aplikacijaun.zrss.si/api/link/ncuzo7y>) in licence,
- » 3.4 Programiranje (<https://aplikacijaun.zrss.si/api/link/rktoqjc>),
- » 4.1 Skrb za varnost naprav (<https://aplikacijaun.zrss.si/api/link/hy2l3ly>),
- » 4.2 Varovanje osebnih podatkov in zasebnosti (<https://aplikacijaun.zrss.si/api/link/2byqenc>),
- » 5.1 Reševanje tehničnih težav (<https://aplikacijaun.zrss.si/api/link/xc2mmi8>),
- » 5.3 Ustvarjalna uporaba (<https://aplikacijaun.zrss.si/api/link/88pa0ty>) digitalne tehnologije.

Z razvijanjem digitalne kompetentnosti pri informatiki prispevamo k oblikovanju dijakov, ki ne bodo zgolj pasivni uporabniki digitalnih tehnologij, temveč jih bodo znali razumeti, nadzorovati in ustvarjalno uporabljati v različnih kontekstih.

TEME, CILJI, STANDARDI ZNANJA

UČINKI RAČUNALNIŠTVA IN INFORMATIKE

OBVEZNO

OPIS TEME

Ta tema spodbuja kritično razmišljanje o tehnološkem napredku in njegovih pozitivnih ter negativnih družbenih posledicah, kar dijake opremi z vedenjem, potrebnim za ozaveščeno in odgovorno delovanje v digitalnem svetu.

DODATNA POJASNILA ZA TEMO

Tema Učinki računalništva in informatike je pomembna pri sklepanju o tem kako bodo tehnologije vplivale na naša življenja v prihodnosti in v katero smer se bo nadaljeval tehnološki razvoj. V resnici tega ni mogoče natančno napovedati, bolje pa si lahko predstavljamo prihodnost (in se nanjo pripravimo), če razumemo, kako se je svet zaradi digitalnih tehnologij spremenil v zadnjih 50 letih. O dijaki pravimo, da so se rodili v digitalni svet, zato »stari« analogni svet poznajo morda le iz pripovedovanja staršev ali medijev. Treba jim je pomagati razumeti, kako, kaj in zakaj so se zgodile takšne korenite spremembe v družbi ter okolju.

UČINKI RAČUNALNIŠTVA IN INFORMATIKE 70

CILJI

Dijak:

○: navaja pozitivne in negativne vplive digitalnih tehnologij na posameznika, družbo (kultura, socialne interakcije, varnostna vprašanja, zakonodajo, etiko) ter okolje,

SC (3.2.5.1)

○: kritično presoja informacije, poglede in potrebe v povezavi s trajnostnim razvojem z vidika naravnega okolja, živih bitij in družbe, pri čemer upošteva različne poglede, pogojene z osebnim, socialnim in kulturnim ozadjem,

SC (2.2.2.1)

○: pojasnjuje nove možnosti, ki jih omogočata vseprisotno komuniciranje in izmenjava podatkov z digitalnimi tehnologijami,

○: sklepa o prihodnjih potencialnih spremembah v družbi in vrednoti mogoče, verjetne in želene scenarije trajnostne prihodnosti.

SC (2.3.1.1)

STANDARDI ZNANJA

Dijak:

- pojasni zakaj smo informacijsko in tehnološko razvita družba, ter pri tem:

- » našteje digitalne tehnologije in izume, ki so v zadnjih 50 letih spremenili svet,
- » **analizira in ovrednoti vpliv digitalnih tehnologij na kakovost življenja,**
- » utemelji prednosti učinkovite rabe digitalnih tehnologij na vseh področjih življenja,
- » opiše nevarnosti in pasti rabe digitalnih tehnologij,
- » predstavi svoj pogled na prihodnje spremembe v družbi kot posledico razvoja digitalne družbe.

TERMINI

◦ umetna inteligenca ◦ veliki jezikovni modeli ◦ generativna UI ◦ digitalni dvojček ◦ 4. in 5. industrijska revolucija ◦ avtomatizacija ◦ internet stvari ◦ navidezni svetovi ◦ kognitivno računalništvo ◦ računalništvo v oblaku

DIDAKTIČNA PRIPOROČILA ZA SKUPINO CILJEV

Tej temi je v prvem letniku namenjeno relativno malo časa, zato naj učitelj kombinira frontalno predavanje s skupinskim delom dijakov. Učitelj izbere primere tehnoloških izumov, ki so korenito spremenili svet. Ob teh primerih naj prikaže, kakšno je bilo nekoč življenje brez teh tehnologij. Pri tem naj omenja pozitivne in negativne vplive na posameznika, družbo ter okolje (poraba energije, odpadki, uničevanje okolja, odtujenost ...). Nato naj učitelj vodi razgovor v smeri novih možnosti, ki jih lahko človeštvo (izboljšana logistika, pametna mesta, izobraževanje, reševanje problemov) in posameznik izkoristita za trajnostno, uspešno in zadovoljujoče življenje. Učitelj naj vsakih deset minut frontalno predavanje prekine z dejavnostjo za dijake z različnimi vnaprej pripravljenimi vprašanji. Na primer:

1. Kako bo tehnologija v prihodnjih desetletjih vplivala na to, kako se učimo, bivamo in delamo?
2. Kakšne so lahko posledice digitalnega prepada med tistimi, ki imajo dostop do naprednih tehnologij, in tistimi, ki ga nimajo?
3. Kako bodo digitalne tehnologije vplivale na zasebnost in varnost posameznikov?
4. Ali lahko tehnologija reši ključne okoljske izzive, s katerimi se soočamo danes?
5. Kako bosta umetna inteligenca in robotika vplivali na etične dileme v družbi?
6. Kako naj družba uravnava uporabo in razvoj tehnologij, da ne bo prišlo do zlorab?
7. Kakšno vlogo igrajo družbeni mediji pri oblikovanju naših političnih in družbenih mnenj?

8. Kakšne etične izzive prinaša tehnologija oblikovanja genov, kot je CRISPR, in kako naj družba obravnava te izzive?
9. Ali bi morali imeti roboti in umetna inteligenca pravice? Če da, kakšne?
10. Kakšen vpliv imata nenehno spremljanje in zbiranje podatkov s strani tehnoloških podjetij na našo avtonomijo?
11. Ali bodo tehnološke inovacije v prihodnosti povečale ali zmanjšale svetovno neenakost?
12. Kakšne bodo posledice, če družbi ne bo uspelo nadzorovati razvoja in uporabe umetne inteligence?

Dijaki naj o vprašanih razpravljajo v skupinah, svoje ugotovitve pa predstavijo ali zapišejo na digitalne table (Padlet, JamBoard ...). Pomembno je, da učitelj da besedo vsakemu dijaku ali da vsaj spodbudi razpravo v skupinah tako, da vsi člani skupine povedo svoje mnenje.

UČINKI RAČUNALNIŠTVA IN INFORMATIKE 210

CILJI

Dijak:

- : vrednoti vpliv razvoja računalniških tehnologij na spremembe v družbi,
- : analizira vplive razširjenih možnosti komuniciranja (lokalno in globalno) na socialne interakcije različnih družbenih institucij (družina, izobraževanje, religija, gospodarstvo, ...),
- : ponazarja etične dileme in zakonske omejitve ter pomembnost varovanja avtorskih pravic na področju digitalnih vsebin.

SC (1.2.2.1 | 1.2.2.2)

STANDARDI ZNANJA

Dijak:

- vrednoti vpliv razvoja računalniških tehnologij na družbo, tako da:

- » opiše različne vplive tehnologij na posamezne skupine ljudi in njihove vsakodnevne dejavnosti,
- » vrednoti kompromise v družbi in iskanje rešitev zaradi globalizacije in avtomatizacije,
- » analizira primere, kjer računalniška tehnologija izboljša, poslabša ali ohranja neenakopraven dostop do informacij in priložnosti.
- » raziskuje mogoče, verjetne in želene vplive digitalnih tehnologij na trajnostnost;

- analizira in pojasni vpliv računalniških tehnologij na socialne interakcije tako da:

- » navede možnosti globalnega in lokalnega sodelovanja,
- » analizira možnosti organiziranja in sodelovanja državljanov pri temah, ki jih zanimajo,
- » pojasni vpliv na povezanost ljudi različnih kultur, potreb ali nazorov,
- » opiše pričakovane spremembe na različnih poklicnih področjih.

- razloži izzive na področju varnosti, zakonodaje in etike tako, da:

- » navede posledice nepooblaščenega rabe digitalnih vsebin,
- » ustrezno navede in citira avtorje uporabljenih virov,
- » primerja različne vrste licenc.

TERMINI

◦ Copyright ◦ CC licence ◦ GPL ◦ slogi citiranja

DIDAKTIČNA PRIPOROČILA ZA SKUPINO CILJEV

Za razvoj kompetence in nabora skupnih ciljev Zamišljanje trajnostnih prihodnosti (SC (2.3.1.1)) lahko učitelj spodbudi učence k razpravi o vplivih digitalne tehnologije na trajnostni razvoj, s poudarkom na razumevanju mogočih prihodnjih scenarijev. Aktivnost lahko poteka kot vodena razprava, kjer dijaki na podlagi znanja in znanstvenih virov ocenjujejo pozitivne in negativne vplive digitalnih tehnologij na okolje, gospodarstvo in družbo.

Dotaknejo se lahko tudi etičnih dilem in zakonskih omejitev (SC (1.2.2.1), SC (1.2.2.2)), ki so potrebne zaradi vpliva umetne inteligence in družbenih omrežij na regulirana področja, kot so volitve in volilne kampanje, spodbujanje čezmerne rabe digitalnih tehnologij (ekonomija pozornosti) idr.

S takšno dejavnostjo bodo dijaki razvijali trajnostno naravnano mišljenje, kar jim bo pomagalo pri oblikovanju stališč in odnosa do odgovorne rabe digitalnih tehnologij.

Učitelj bi dejavnost lahko izvedel v naslednjih korakih:

V uvodu bi dijakom predstavil cilje trajnostnega razvoja in poudaril, kako digitalna tehnologija vpliva na trajnostnost (npr. vpliv na okoljske vire, energetska poraba modelov umetne inteligence, podatkovnih centrov ... digitalno pravičnost, dostop do informacij, recikliranje e-odpadkov itd.).

Nato bi se dijaki razdelili na manjše skupine, v katerih bi vsaka skupina razpravljala o enem vidiku vpliva digitalne tehnologije na trajnostnost (npr. digitalna tehnologija in okolje, digitalna tehnologija in družbena pravičnost, digitalna tehnologija in gospodarstvo).

Vsaka skupina bi oblikovala dva scenarija:

- » mogoči scenarij (verjeten razvoj ob trenutnih trendih),
- » želeni scenarij (kako bi morala digitalna tehnologija podpirati trajnostnost v prihodnosti).

Na koncu bi sledila skupna razprava, kjer bi skupine predstavile svoje scenarije, dijaki pa bi ocenili, kateri scenariji so bili najbolj trajnostni in zakaj. Učitelj bi vodil razpravo in spodbujal kritično razmišljanje ter povezoval ugotovitve s konceptom trajnostnega razvoja.

Dijakom je treba pokazati rabo orodij za citiranje in navajanje virov (ter izdelavo kazala virov), ki so vgrajeni v urejevalnike besedil in spletna orodja, kot je na primer Zotero. Dijakom se zaradi uporabe teh orodij ni treba ukvarjati z značilnostjo posameznega sloga citiranja (na primer APA, CMOS, IEEE ...), ampak v obrazec samo vpišejo ustrezne podatke in izberejo slog. Vse ostalo izpiše/citira/navede in uredi urejevalnik besedil.

Pri licencah je treba predstaviti značilnosti in razlike med licenco CC in licenco Copyright. Učence se spodbuja, da na svoja gradiva umestijo licenco CC, pri tem pa izberejo ustrezne lastnosti (CC BY, SA, NC, ND). Dodatna navodila in opise najdejo na spletni strani Ustvarjalnagmajna.si in v okvirju digitalnih kompetenc DigComp 2.2 za kompetenco 3.3 Avtorske pravice in licence (<https://aplikacijaun.zrss.si/api/link/ncuzo7y>) (SC (4.3.3.1)).

VPLIV NA NARAVO IN VSEBINO POKLICEV 70

CILJI

Dijak:

- : predstavlja inženirska področja, ki so povezana z računalništvom in informatiko,
- : analizira povezanost posameznih inženirskih področij z računalništvom in informatiko,
- : vrednoti, kako poznavanje področja računalništva in informatike pripomore k lažjemu prilagajanju družbe na potrebe sedanjosti in prihodnosti.

STANDARDI ZNANJA

Dijak:

- s pomočjo primerov iz prakse razloži povezanost inženirskih področij (ki so povezana z računalništvom in informatiko) z izzivi in potrebami družbe tako da:

- » našteje posamezna inženirska področja, njihove značilnosti in poklice s teh področij,
- » izbere primarno inženirsko področje za podano tehnološko rešitev/izdelek/storitev,
- » navede ostala inženirska področja, ki dopolnjujejo tehnološko rešitev/izdelek/storitev,
- » navede izzive družbe in v povezavi z njimi inženirska področja, ki jih pomagajo reševati.

TERMINI

○ računalniška omrežja ○ robotika ○ telekomunikacije ○ avtomatika ○ elektronika ○ močnostna elektrotehnika ○ strojništvo ○ gradbeništvo

DIDAKTIČNA PRIPOROČILA ZA SKUPINO CILJEV

Pri tej skupini ciljev želimo, da dijaki:

- » spoznajo različna inženirska področja, ki vključujejo računalništvo in informatiko.
- » razumejo, kako se področja prepletajo pri razvoju tehnologij in reševanju družbenih izzivov.
- » kritično ocenijo, katera področja jih zanimajo in kako lahko svoje interese povežejo s študijsko izbiro po koncu srednje šole.

Učitelj lahko na začetku sklopa (dveh ur) prične z mehčanjem stereotipov o računalništvu in informatiki ter s predstavitvijo interdisciplinarnosti različnih področij. Z dijaki lahko naredi miselni eksperiment (10 min.) »**Kdo je računalnikar oz. informatik?**«, tako da:

- » dijake vpraša, kakšno podobo imajo o računalništvu in informatiki,

- » dijaki skupinsko zbirajo ključne besede, asociacije in jih pišejo na tablo (npr. programer, gamer, umetna inteligenca, heker ...),
- » učitelj pokaže nekaj slik poklicev (mehatronik, biomedicinski inženir, avtomatizacija v industriji, elektroinženir, telekomunikacijski inženir) in vpraša: »Ali ti poklici vključujejo računalništvo?«

Nato razdeli dijake v skupine po 3–4 in vsaki skupini dodeli en sodoben tehnološki izdelek ali rešitev, npr.:

- » pametna ura,
- » avtonomni avtomobil,
- » industrijski robot,
- » telekomunikacijsko omrežje (5G),
- » medicinska diagnostika z UI.

Skupine pri tej dejavnosti razmišljajo (15 min.):

- » Katere inženirske smeri sodelujejo pri razvoju tega izdelka/rešitve?
- » Kako se pri tem uporabljata računalništvo in informatika?
- » Kateri poklici pri tem sodelujejo?

Skupine po razpravi na kratko predstavijo svoje ugotovitve.

Nato lahko učitelj prikaže povezavo računalništva in informatike z različnimi inženirskimi področji na konkretnih primerih (20 min.):

- » **strojništvo** (avtomobilska industrija, 3D-modeliranje, programska oprema CAD),
- » **mehatronika** (pametni stroji, industrijski roboti, avtomatizacija),
- » **elektrotehnika** (razvoj čipov, internet stvari, komunikacijski sistemi),
- » **biomedicinska tehnika** (medicinska diagnostika, računalniška tomografija),
- » **telekomunikacije** (mobilna omrežja, internetna infrastruktura, optična vlakna).

Učitelj lahko dijake spodbuja, da razmišljajo o interdisciplinarnosti področij in o tem, kje bi lahko v prihodnosti sami našli svoje mesto.

V drugi šolski uri se dijaki razdelijo v skupine in dobijo nalogo (30 min.) z naslednjimi navodili:

- » Predstavljajte si, da ste ekipa strokovnjakov in razvijate nov tehnološki izdelek ali rešitev za prihodnost (npr. pametno mesto, zelena energija, digitalno zdravstvo).
- » Katera inženirska področja bodo sodelovala pri tem projektu?
- » Kako bosta računalništvo in informatika igrala ključno vlogo?
- » Kakšni poklici bodo vključeni?

Skupine oblikujejo kratko predstavitev (lahko s plakatom ali z digitalnim orodjem) in na koncu predstavijo svoje ideje.

Na koncu sledita refleksija in razmislek (15 min.) o lastni karierni poti, pri čemer lahko učitelj zastavi naslednja vprašanja:

- » Katera področja so vas najbolj pritegnila?
- » Ste ugotovili, da vas zanimajo poklici, na katere prej niste pomislili?
- » Kako boste raziskovali naprej, preden izberete študijsko pot?

Dijaki lahko delijo svoja razmišljanja, učitelj pa poudari, da se računalništvo in informatika uporabljata na širokem spektru področij in da je ključno raziskovati možnosti zunaj klasičnih izbir.

Z opisano dejavnostjo bodo dijaki dobili širšo sliko o tem, kako se računalništvo in informatika prepletata z različnimi inženirskimi smermi, kar jim lahko pomaga pri informiranem odločanju o nadaljnjem izobraževanju.





RAČUNALNIŠKI SISTEMI

OBVEZNO

OPIS TEME

Dijaki spoznavajo zgradbo računalnikov tako z vidika strojne in programske opreme kot tudi z vidika podatkov. Spoznavajo, kako se podatki digitalizirajo, zapisujejo in shranjujejo. Poudarek je na razumevanju, kako strojna in programska oprema medsebojno komunicirata in procesirata podatke.

Dijaki razumejo, zakaj razumevanje področja računalništva in informatike pripomore k lažjemu prilagajanju družbe na potrebe sedanjosti in prihodnosti.

DODATNA POJASNILA ZA TEMO

Večina dijakov v prvem letniku nima specifičnega predznanja o zgradbi in delovanju računalnika. Zato pri obravnavi teme uporabljamo analogije in se navezujemo na njihovo predznanje iz vsakdanjega življenja.

VON NEUMANNOV MODEL 70

CILJI

Dijak:

- : primerja vlogo posameznih elementov von Neumannove arhitekture računalnika,
- : interpretira korake izvajanja ukazov v CPE,
- : prepozna tehnične težave pri delu z napravami, jih zna opisati in določene odpravljati (DigComp: 5.1).
-  (4.5.1.1)

STANDARDI ZNANJA

Dijak:

- ponazori zgradbo in delovanje von Neumannovega modela računalnika tako da:

- » diagramsko prikaže von Neumannov model,
- » opiše vlogo posamezne komponente (centralno procesna enota - CPE, pomnilnik, vhodno-izhodne (VI) naprave) pri izvajanju nalog,

- » opiše bistvene tehnične lastnosti posamezne komponente;
- pojasni razlike med računskimi stroji, računalniki in mikrokrmilniki tako da:
 - » opiše razlike v zgradbi naprav,
 - » primerja uporabnost teh vrst naprav.
- razloži pojme: podatek, program in ukaz, tako da:
 - » opiše kaj so podatki, ukazi in programi ter kako se razlikujejo med seboj,
 - » navede primer uporabe vsakega koncepta v računalniškem okolju.

TERMINI

- strojna oprema ◦ delovni pomnilnik ◦ trajni pomnilnik ◦ von Neumannov model računalnika ◦ centralna procesna enota ◦ vhodno-izhodna enota

DIDAKTIČNA PRIPOROČILA ZA SKUPINO CILJEV

S pojmom »računski stroji« mislimo na naprave (zgodovinsko gledano), ki so bile zasnovane za izvajanje matematičnih operacij (abakus, mehanska računalna /Pascalina, Babbageev analitični stroj/, ENIAC, Colossus, EDSAC ...) in niso bile zgrajene po von Neumannovem modelu računalnika.

Ko načrtujemo učne ure, si lahko za boljše razumevanje novih pojmov in konceptov pomagamo s prosto dostopnimi spletnimi viri, kot so:

- » <http://vidra.si>,
- » <https://edu.qcglobal.org/en/computerbasics>,
- » <https://aplikacijaun.zrss.si/api/link/xyonii9>,
- » <https://www.csunplugged.org/en>.

Dober način, kako začeti s to temo, najdemo na strani *code.org* (<https://aplikacijaun.zrss.si/api/link/n1a78ku>). Na strani najdemo podroben opis aktivnosti, učne načrte za izvedbo posamezne ure in gradiva za dijake.

Kratek opis mogoče izvedbe uvoda v temo:

Z dijaki začnemo pogovor o vsakodnevnih problemih in kako jih rešujejo. Nato jih usmerimo na razmišljanje o tem, kakšna orodja pri tem uporabljajo. Iz tega izpeljemo, da danes veliko problemov rešujemo s pomočjo računalnika. Dijake spodbudimo, da razmišljajo, kateri so ti problemi. Tukaj ni poudarek na reševanju problemov, ampak želimo, da dijaki pridejo do zaključka, da jim računalnik danes služi kot orodje za reševanje mnogih vsakodnevnih problemov.

Nato se z dijaki pogovarjamo, kaj je računalnik in kako deluje ter kako je sestavljen, da lahko tako deluje.



V prvem letniku se posvetimo predvsem opisu delovanja glavnih komponent von Neumannove arhitekture in njihovi medsebojni povezanosti (vhodno-izhodne enote, CPE, pomnilnik). Poudarek pri tem damo na opis koncepta enotnega pomnilnika za pomnjenje tako ukazov kot podatkov.

Naštujemo vrste trajnih pomnilnikov: bliskovni pomnilniki (SSD, USB, SD), magnetni pomnilniki (trdi disk – HDD, tračne enote), optični pomnilniki (CD, DVD).

Da zadržimo pozornost dijakov na temi, razlago večkrat prekinemo z vprašanji ali ogledom poučnih videoposnetkov (npr. <https://www.youtube.com/watch?v=AkFi90IZmXA>, Computer Basics: Inside a Computer (gcfglobal.org (<https://edu.gcfglobal.org/en/>))). Za ideje si pri oblikovanju pravih vprašanj lahko pomagamo tudi z jezikovnimi modeli, ki nam znajo hitro sestaviti vprašanja, primerna za obravnavano temo.

Izvedemo lahko dejavnost, v kateri dijaki predstavljajo delovanje posameznih komponent von Neumannovega modela (<https://aplakacijaun.zrss.si/api/link/qk48wpa>, <https://cse4k12.blogspot.com/search/label/ALU>).

PRIPOROČENI NAČINI IZKAZOVANJA ZNANJA

Ustno ali pisno izkazovanje znanja.

VON NEUMANNOV MODEL 210

CILJI

Dijak:

○: analizira namen komponent strojne ter programske opreme računalniških sistemov in njihovo medsebojno delovanje,

○: prepoznava in rešuje preproste probleme s strojno in programsko opremo (DigComp 5.1).

SC (4.5.1.1)

STANDARDI ZNANJA

Dijak:

- razloži način delovanja von Neumannovega modela in pri tem:

» opiše namen, delovanje in medsebojno povezanost komponent:

» - CPE (aritmetične enote, krmilne enote, ukaznega registra, programskega števca),

» - vodil,

» - pomnilnika in

» - vhodno-izhodnih enot

» ter kako te vplivajo na zmogljivost računalnika;

- pojasni hierarhijo pomnilnikov (vključno s predpomnilnikom) in njihov vpliv na delovanje računalnika;

- pojasni razlike v zgradbi, delovanju in namenu mikroprocesorjev, mikrokrmilnikov in računalnikov;

- prepozna enostavne napake v delovanju strojne in programske opreme ter jih odpravi. SC (4.5.1.1)

TERMINI

◦ programski števec ◦ vodilo ◦ predpomnilnik

DIDAKTIČNA PRIPOROČILA ZA SKUPINO CILJEV

Preden začnemo poučevati nova znanja, z dijaki ponovimo predelane vsebine prvega letnika in osvežimo morebitna manjkajoča znanja. Pri tem si pomagamo z didaktičnimi priporočili za obveznih 70 ur. Ta znanja so temelj razumevanja delovanja računalniških sistemov. Nato jih lahko nadgradimo z zahtevnejšimi vsebinami, ki dijakom omogočajo poglobljeno razumevanje delovanja računalnika in vloge posameznih komponent računalnika.

Za razlago delovanja programskega števca in ukaznega registra lahko pred razlago pogledamo izobraževalni video: <https://www.youtube.com/watch?v=ccf9ngGlb8c> in nato z dijaki razpravljamo o videnem.

Ko dijakom razlagamo razlike med mikroprocesorjem, mikrokrmilnikom in računalnikom, lahko uporabimo analogije iz življenja, ki so jim blizu in jih razumejo, na primer:

Mikroprocesor (CPU)

»Mikroprocesor je kot kuhar, ki zna odlično kuhati, a nima sestavin (pomnilnika) in kuhinjskih pripomočkov (V/I). Potrebuje pomočnike.«

Mikrokrmilnik (MCU)

»Mikrokrmilnik je kot kuhar, ki ima vse, kar potrebuje v nahrbtniku – kuhinjo, sestavine in recepte. Gre na teren in sam skuha jed.«

Računalnik

»Računalnik je kot celotna restavracija – kuhar (procesor), shramba (pomnilnik), jedilnica (zaslon), natakarji (vhodno-izhodne naprave) in jedilni list (programi).«



SISTEMSKA PROGRAMSKA OPREMA 70

CILJI

Dijak:

○: analizira vlogo operacijskega sistema in ostale programske opreme.

STANDARDI ZNANJA

Dijak:

- analizira vlogo operacijskega sistema, tako da:

- » raziskuje, kako operacijski sistem upravlja vire in jih nudi uporabniški programski opremi v uporabo,
- » za procese vrednoti dodeljeno količino virov (količino pomnilnika, procesorskega časa).

TERMINI

○ operacijski sistem ○ program ○ aplikacija

DIDAKTIČNA PRIPOROČILA ZA SKUPINO CILJEV

Za boljšo ponazoritev in večjo angažiranost dijakov pri obravnavi operacijskih sistemov vključimo izobraževalne videoposnetke, ki s pomočjo vizualnih animacij in konkretnih primerov pojasnjujejo, kako delujejo procesi, upravljanje pomnilnika in uporabniški vmesniki. Nekaj predlogov izobraževalnih videoposnetkov:

» <https://aplikacijaun.zrss.si/api/link/nylv2yq>

» <https://aplikacijaun.zrss.si/api/link/k8v7lu1>

» <https://aplikacijaun.zrss.si/api/link/fc3rom7>

Za boljše razumevanje vsebin na posnetkih priporočamo vklop strojnega prevajanja v slovenščino.

SISTEMSKA PROGRAMSKA OPREMA 210

CILJI

Dijak:

○: analizira, kako sistemska programska oprema povezuje uporabniške aplikacije in strojno opremo.

STANDARDI ZNANJA

Dijak:

- analizira vlogo operacijskega sistema, tako da:

- » **našteje in analizira sistemske vire, ki jih upravlja operacijski sistem,**
- » opredeli pojem procesa,
- » analizira, kako operacijski sistem upravlja sistemske vire in jih dodeljuje procesom, tako da opiše:
- » - dodeljevanje sistemskih virov procesom (CPE, pomnilniški prostor, vhodno-izhodne enote),
- » - upravljanje sočasnega dostopa do virov,
- » - optimiziranje in nadziranje porabe sistemskih virov,
- » **za posamezen proces razbere vire, ki so mu bili dodeljeni, in njihovo količino.**

TERMINI

◦ programska oprema ◦ sistemska programska oprema ◦ operacijski sistem ◦ sistemski viri ◦ procesi

DIDAKTIČNA PRIPOROČILA ZA SKUPINO CILJEV

Dijak naj spozna, da vsakič, ko zaženemo program ali aplikacijo, se le-ta izvaja v ločenem prostoru delovnega pomnilnika.

Za boljše razumevanje delovanja operacijskega sistema je priporočljivo uporabiti didaktična orodja, na primer <https://teach-sim.com>. Ta izobraževalna programska oprema je zasnovana za podporo računalniškemu izobraževanju. Simulira delovanje procesorja in operacijskega sistema ter je prosto dostopna.



DIGITALIZACIJA PODATKOV 70

CILJI

Dijak:

- : zapisuje celoštevilsko vrednost v desetiškem, dvojiškem in šestnajstiškem zapisu,
- : računa potrebno število bitov za predstavitev informacije,
- : primerja, kako lahko v bitnem zapisu predstavimo besedilo in slike.

STANDARDI ZNANJA

Dijak:

- » pojasni, kaj je bit, zlog in niz bitov,
- » uporablja predpone za zapis količine podatkov,
- » zapiše celo število, znak ali barvno piko z bitnim zapisom,
- » niz bitov pretvori v število, besedilo ali sliko,
- » analizira odvisnost med številom bitov v zapisu in številom različnih zapisov, ki jih to število bitov omogoča,
- » izračuna število bitov potrebnih za zapis različnih množic podatkov,
- » ustvari kodno tabelo za dane podatke ter utemelji izbiro kodiranja,
- » izračuna količino podatkov za nestisnjen zapis slike.

TERMINI

○ količina informacije ○ bit ○ zlog ○ številski sistemi

DIDAKTIČNA PRIPOROČILA ZA SKUPINO CILJEV

Predlagamo, da pri tej temi uporabimo dejavnosti, ki so opisane na spletnih straneh VIDRA – računalništvo brez računalnika (<http://vidra.si/index.html>) in na spletnih straneh CS Unplugged (<https://www.csunplugged.org>):

- » Zapis števil (<http://vidra.si/dvojski-zapis-stevil>).
- » Zapis znakov (<http://vidra.si/zapis-in-prenos-besedil/>, <http://vidra.si/zapis-in-prenos-besedil/>, <http://vidra.si/stiskanje-besedil>).

- » Zapis slik (<http://vidra.si/stiskanje-slik>).

Narišimo sliko in jo shranimo v formatu BMP, nato jo odprimo z orodjem, ki nam prikaže heksadecimalno kodo.

- » Količina informacije in kodiranje (<https://aplikacijaun.zrss.si/api/link/vp3plac>).

Tu dijaki lahko rešujejo naloge, kjer morajo določiti dolžino bitnega zapisa, s katerim bodo zakodirali sliko ali besedilo. Na podlagi tega podatka ustvarijo kodno tabelo.

- » Stiskanje podatkov (<http://vidra.si/krajsi-zapis-besedila/>).

Aktivnost lahko izvedemo tako, da dijaki narišejo sliko in jo shranijo v različnih formatih. Nato skupaj primerjamo velikosti dobljenih datotek in razmišljamo ter razpravljamo, zakaj so velikosti datotek različne.

Pri izračunu števila bitov za zapis različnih množic podatkov pripravimo primere za:

- » male črke abecede ali kak drug nabor znakov,
- » met kocke,
- » različne barvne palete.

DIGITALIZACIJA PODATKOV 210

CILJI

Dijak:

- opisuje, kako zvok in video predstavimo v dvojiškem zapisu,
- decimalna števila predstavlja v zapisu s plavajočo vejico,
- primerja različne oblike stiskanja podatkov.

STANDARDI ZNANJA

Dijak:

- opiše in razloži, kako pretvorimo zvok in video v dvojiški zapis;
- pretvori decimalno število v dvojiški zapis;
- izračuna količino podatkov za nestisnjen zapis slike in zvoka:
 - » za podan primer slike, besedila, zvoka ali videa:
 - » - primerja različne oblike stiskanja podatkov,
 - » - izbere najprimernejšega in
 - » - argumentira svoj izbor,
 - » za konkreten primer podatka primerja različne oblike stiskanja in argumentira izbor najustrežnejšega.

TERMINI

- količina informacije ○ bit ○ bajt ○ zapis s plavajočo vejico ○ stiskanje podatkov

DIDAKTIČNA PRIPOROČILA ZA SKUPINO CILJEV

Razlago digitalizacije zvoka lahko začnemo z zvočnim posnetkom in nato predstavimo nekaj dejstev o zvoku, na primer:

- » Zvok je valovanje s frekvencami v slišnem območju človeškega ušesa, to je med 20 in 20.000 Hz. Zvok z nižjimi frekvencami imenujemo infrazvok, zvok z višjimi frekvencami pa ultrazvok. Zvok, ki ga sestavlja le ena sinusoida (ena frekvenca), imenujemo ton.
- » Nato lahko uporabimo katero od aplikacij, na primer <https://academo.org/music>, ki prikazuje valovanje zvoka v realnem času.



- » Potem razložimo, da računalnik ne razume zvoka, ampak številke in da je digitalizacija pretvorba neprekinjenega (analognega) signala v diskretno zaporedje števil.
- » Pretvorba se izvede z vzorčenjem, kjer se vzorci zapisujejo z različnim številom bitov. Če imamo možnost, dijakom predvajamo posnetek, kjer je vsak vzorec zapisan z 8 biti, in posnetek s 16 biti. Primerjamo kakovost obeh posnetkov.
- » Vzorčenje in zapis zvoka lahko razložimo tudi ob ogledu izobraževalnega videoposnetka, kot je na primer <https://www.youtube.com/watch?v=q3tfly9mKhY>.

Za razlago zapisa števila s plavajočo vejico nam lahko pomagajo orodja, kot sta

<https://aplikacijaun.zrss.si/api/link/ai2cjjq> in <https://evanw.github.io/float-toy/>, ki vizualno prikažeta, kako se decimalno število pretvori v digitalno, in prikažeta mantiso, eksponent in predznak.



PRENOS IN PRAVILNOST ZAPISA PODATKOV 210

CILJI

Dijak:

- : raziskuje problematiko spreminjanja podatkov in pomembnost njihove nespremenljivosti,
- : uporablja preproste načine in metode za preverjanje pravilnosti zapisa podatkov.

STANDARDI ZNANJA

Dijak:

- » navede nekaj scenarijev, kjer se podatki lahko spreminjajo in zanje opiše načine zaznave morebitnih sprememb,
- » na danih primerih uporabi preproste metode za zaznavo sprememb.

TERMINI

- celovitost
- paritetni bit

DIDAKTIČNA PRIPOROČILA ZA SKUPINO CILJEV

Za boljše razumevanje pomena celovitosti podatkov in načinov za zagotavljanje le-tega lahko izvedemo dejavnosti, predstavljene na straneh:

- » <http://vidra.si/pokvarjeni-bit>,
- » <https://aplikacijaun.zrss.si/api/link/kw1p96a>,
- » <https://aplikacijaun.zrss.si/api/link/aq76mj8>.





PODATKI IN ANALIZA

OBVEZNO

OPIS TEME

Dijaki spoznavajo ključne vidike zbiranja, organiziranja, obdelave in analize podatkov. Z interpretacijo podatkov in predvidevanjem dogodkov na osnovi te interpretacije, se učijo razumevati svet okoli sebe.

DODATNA POJASNILA ZA TEMO

Postopke zbiranja, organiziranja, obdelave in analize podatkov je mogoče izvesti z različnimi oblikami podatkov in z zelo različno programsko pa tudi strojno opremo. Pri tem razvijamo t. i. podatkovno pismenost.

Predlagamo, da:

- » dijaki v obveznem delu (70 ur) podatke obdelujejo v preglednicah (vpisovanje podatkov, uporaba formul in funkcij, prikazovanje podatkov z ustreznimi grafikoni ...) in spletnih orodjih za vizualizacijo,
- » naj dijaki v razširjenem delu (210 ur) spoznajo osnove relacijskih podatkovnih baz in s strojnim učenjem ter vizualizacijo odkrivajo vzorce v podatkih in tako poskušajo razumeti zgodbo, ki jo podatki »pripovedujejo«.

Predlagamo, da se ta skupina ciljev obravnava pri urah, kjer lahko vsak dijak dela na svojem računalniku.

Namen teme Podatki in analiza nikakor ni samo učenje uporabe orodij (npr. Google Sheets, SQLite, Excel, Access, Orange ...), ampak predvsem razvijanje veščin sistematičnega iskanja, razumevanja, vrednotenja in uporabe podatkov pri reševanju problemov. Pri dejavnostih spodbujamo dijake, da se sprašujejo tudi, ZAKAJ obdelujemo podatke in ne le KAKO uporabljati določena orodja za delo s podatki.

DIDAKTIČNA PRIPOROČILA ZA TEMO

Okvirne vsebine za relacijske podatkovne baze (predlagamo 16 ur)

Predlagamo, da pouk izvedete tako, da snov in teorijo razlagate ob vodeni gradnji podatkovnega modela in nato tudi podatkovne baze. Pouk lahko razdelite na naslednje korake:

- » Modeliranje podatkovnega modela in grajenje konceptualnega modela (model entiteta-razmerje oz. ER-model)
- » Razložite pojme, kot so podatkovni model, konceptualni model, entiteta, atribut in relacija.
- » S pomočjo primerov iz realnega sveta (npr. podatki o učencih) naj dijaki oblikujejo konceptualni model (ER-model).



- » Seznanitev z orodjem za delo z bazami podatkov
 - » Dijaki spoznajo osnovne funkcije in uporabniški vmesnik orodja za delo z bazami podatkov.
 - » Dijaki po demonstraciji ustvarijo preprosto bazo podatkov za simulacijo praktičnega primera (v katero bodo kasneje preslikali prej načrtovani ER-model).
- » Izdelava logičnega modela
 - » Razložite pojme, kot so logični model, tabela (seznam) in zapis v tabeli (seznamu), podatek in podatkovni tipi, ključi.
 - » Praktično preslikajte entitete iz ER-modela v tabele z ustreznimi ključi in podatkovnimi tipi.
 - » Dijaki naj izvajajo vaje normalizacije, da se naučijo zmanjšati redundanco podatkov.
- » Zbiranje in shranjevanje podatkov
 - » Dijaki po demonstraciji uvozijo ali sami vnesejo podatke v tabele v podatkovni bazi. (Podatkov naj bo dovolj, da se bo nad njimi dalo izvajati smiselna povpraševanja.)
- » Prikazovanje, preoblikovanje in sklepanje
 - » Naučite osnove jezika SQL (osredotočite se na stavek SELECT, preostale /INSERT, UPDATE, DELETE/ pa pokažite po svoji presoji).
 - » Dijaki naj zapisujejo in izvajajo poizvedbe za različne scenarije nad eno in dvema tabelama (uporaba tudi tujih ključev).

Okvirne vsebine za strojno učenje in vizualizacijo (predlagamo 24 ur)

- » Uvod v algoritme strojnega učenja
 - » Razložite osnove strojnega učenja, razliko med nadzorovanim in nenadzorovanim učenjem ter aplikacije v resničnem svetu.
 - » Dijaki naj raziskujejo preproste primere, kot je napovedovanje ocen na podlagi podatkov.
- » Zajem in predobdelava podatkov
 - » Demonstrirajte, kako uvoziti podatke v izbrani program (npr. CSV ali Excel).
 - » Dijaki naj vadijo čiščenje in urejanje podatkov (odstranjevanje manjkajočih vrednosti, normalizacija).
- » Strojno učenje: nadzorovano in nenadzorovano učenje
 - » Razložite algoritme, kot so Naivni Bayesov klasifikator (Naive Bayes), k-najbližjih sosedov (k-NN), Metoda s podpornimi vektorji (SVM), odločitvena drevesa (za klasifikacijo) ter gručenje (npr. gručenje s K središči (K-means).
 - » Dijaki naj sami izvedejo analize s pomočjo grafičnega vmesnika.

- » Koncept vložitve (embedding)
 - » Razložite vložitev podatkov in njihov pomen za obdelavo slikovnih in besedilnih podatkov.
 - » Dijaki naj obravnavajo primer enostavne obdelave slik ali besedil.
- » Vrednotenje modelov
 - » Pojasnite kriterije uspešnosti (npr. točnost, natančnost, priklic, F1-mera) in pomen prečnega preverjanja (cross-validation).
 - » Dijaki naj ocenjujejo modele na primerih, da pridobijo občutek za interpretacijo rezultatov.

ZBIRANJE IN SHRANJEVANJE 70

CILJI

Dijak:

- : pridobiva podatke na različne načine,
- : shranjuje podatke v datoteke ustreznega formata na primerno in hierarhično ter logično smiselno lokacijo,
 - SC (4.1.3.1)
- : interpretira vlogo datoteke in pomen njenih metapodatkov.

STANDARDI ZNANJA

Dijak:

- pridobi različne vrste podatkov na ustrezne in različne načine;
- shrani podatke, tako da:
 - » ustvari pregleden in logičen sistem map,
 - » datoteke ustrezno poimenuje,
 - » izbere ustrezne formate datoteke,
 - » izbere ustrezno mapo,
 - » izvrši ukaz shranjevanja in preveri, ali je vsebina shranjena v pravi obliki na želenem mestu;
- poišče iskano vsebino, tako da:
 - » išče iskano datoteko v strukturi map ali



- » išče z vgrajenimi iskalniki z določanjem različnih parametrov: ime, vrsta datoteke, čas nastanka, velikost, vsebina.

TERMINI

- entiteta ◦ atribut ◦ podatek ◦ informacija ◦ spletni vprašalnik ◦ podatkovna zbirka ◦ preglednica
- datoteka ◦ format ◦ imenik ◦ mapa ◦ pogon ◦ datotečna struktura ◦ oblak

DIDAKTIČNA PRIPOROČILA ZA SKUPINO CILJEV

Predlagamo, da učitelj pokaže dijakom različne možnosti pri pridobivanju podatkov. Ob tem uporabljajo načinu ustrezna orodja (za zbiranje podatkov) oziroma algoritme/postopke:

- » Iz obstoječih (digitalnih) zbirk. Uporabimo lahko dostopne avtentične podatke (npr. športni izidi, podatki o gledanosti filmov v tekočem letu, podatki o popularni glasbi ...), druge zanimive aktualne podatke (vremenski podatki, potresi ...) ali podatke, povezane s snovjo drugih predmetov (npr. družbenoekonomski podatki svetovne banke).

Nekaj virov prosto dostopnih podatkov:

- » Odprti podatki Slovenije: <https://podatki.gov.si>,

- » podatki EU: <https://ec.europa.eu/eurostat/data/database>,

- » Kaggle: <https://www.kaggle.com/datasets> (<https://www.kaggle.com/datasets>),

- » okolje Orange vsebuje gradnik Zbirke podatkov / Datasets s podatki, ki so že pripravljeni za uvoz in večinoma namenjeni poučevanju.

- » Iz obstoječega »knjižnega/analognega« gradiva (pri tem se lahko pogovarjamo o pojmi, kot so optično odčitavanje knjig; OCR; digitalizacija knjižnega/slikovnega gradiva).

V prvem letniku oz. v obsegu obveznih 70 ur se ne lotimo zbiranja podatkov s senzorji.

ZBIRANJE IN SHRANJEVANJE 210

CILJI

Dijak:

- : podatke pridobiva na različne načine in pri tem uporablja ustrezna orodja ter algoritme oz. postopke,
- : opredeljuje temeljne gradnike relacijske podatkovne baze (tabela, poizvedba, polje, podatkovni tipi, ključ, zapis),
- : opisuje in vrednoti načine zagotavljanja celovitosti in smiselnosti shranjevanih podatkov,
- : opisuje vpliv pridobivanja in shranjevanja podatkov na stroške, hitrost, zanesljivost, dostopnost, varovanje zasebnosti in njihovo celovitost,
- : za podane primere izdeluje relacijske podatkovne baze.

STANDARDI ZNANJA

Dijak:

Dijak:

- **pojasni različne načine pridobivanja podatkov in v praksi uporabi vsaj dva;**
- pridobljene podatke shrani in ob tem:
 - » uporabi in utemelji izbiro programske opreme,
 - » izbere in utemelji izbiro formata datoteke;
- pojasni, kako način pridobivanja podatkov in izbira orodja vplivata na količino ter kakovost pridobljenih podatkov;
- utemelji razloge za avtomatizirano pridobivanje podatkov;
- prepozna in opiše primere, v katerih pridobivanje podatkov posega v zasebnost, in pojasni, zakaj;
- **opredeli relacijsko podatkovno bazo in temeljne sestavine tabele (podatek, atribut, polje, podatkovni tip, ključ, zapis);**
- **opiše lastnosti osnovnih tipov podatkov (znakovni, številski, datum, logični, besedilo) in jih ponazori s primeri;**
- razloži pomen primarnega in tujega ključa;
- zagotovi in utemelji celovitost ter smiselnost shranjevanih podatkov, tako da podatke opremi z ustreznimi medsebojno povezanimi spremljajočimi podatki;
- omeji dostopnost in zagotovi varovanje zasebnosti;



- izdelava relacijski model podatkovne baze, tako da:

- » ustvari tabele,
- » določi polja in tipe podatkov,
- » določi primarne in tuje ključe,
- » poveže tabele z relacijami,
- » odpravi podvajanja in onemogoči nekonsistentnost baze,
- » vpiše podatke;

- razlikuje med tabelo in preglednico.

TERMINI

◦ senzor ◦ podatkovna baza ◦ podatkovni tip ◦ dostopnost podatkov ◦ zgradba datoteke ◦ metapodatki

DIDAKTIČNA PRIPOROČILA ZA SKUPINO CILJEV

Predlagamo, da dijaki podatke pridobivajo (zbirajo) na različne načine. Ob tem poleg že poznanih orodij in algoritmov (oz. postopkov za zbiranje podatkov) uporabljajo še naslednje možnosti:

- » senzorje za zaznavanje temperature, vlage, gibanja, svetlobe, dotika, razdalje idr.,
- » periodično zajemanje podatkov s spletnih strani.

Razložimo, da se v senzorju običajno merjeni podatki preoblikujejo iz analogne v digitalno vrednost, kot je to na primer pri senzorju za merjenje temperature, kjer senzor opravi naslednje korake:

- » zaznavanje temperature (fizični ali električni odziv na temperaturo),
- » pretvorba (sprememba signala npr. napetosti ali upornosti v uporabne podatke),
- » prenos podatkov (pošiljanje podatkov v napravo prikaz – mikrokrmilnik ali zaslon).

Pokažemo, da lahko do podatkov pridemo tudi preko spletnih strani, kjer se zajeti podatki periodično objavljajo. Tako lahko na primer podatek za temperaturo zraka odčitamo vsako uro s spletne strani meteorološkega zavoda in ga shranimo na svoj računalnik. Dijakom pokažemo, kako takšno pridobivanje podatkov iz spletne strani avtomatiziramo.

Dijakom razložimo, da izbira orodja in količina zajetih podatkov vplivata na kakovost, tako da prikažemo primer:

- » Primer, kjer količina zajetih podatkov vpliva na kakovost pridobljenih podatkov: Zanima nas, kako niha temperatura v učilnici čez celoten dan. To lahko storimo tako, da s senzorjem zajamemo trenutno temperaturo vsake štiri ure, lahko pa zajemamo temperaturo vsakih 30 minut ali vsako sekundo. V prvem primeru bomo verjetno hitro zgrešili morebitne temperaturne ekstreme, saj je časovni interval zelo dolg. V tretjem primeru, ko merimo temperaturo vsako sekundo, pa se izkaže, da sta zajemanje in obdelava

zajetih podatkov po nepotrebnem prezahtevna in »predraga«. Prepogosto zajemanje temperature ne prinaša nobenih prednosti, ampak samo povečuje količino podatkov in s tem zahtevnost njihove obdelave.

- » Primer, kjer izbira orodja vpliva na kakovost pridobljenih podatkov: Upoštevati je treba omejitve orodja (senzorja) za zajemanje podatkov. Pri merjenju temperature s senzorji Vernier dobimo boljše podatke kot pri merjenju temperature z micro:bitom. Micro:bit namreč temperaturo okolice ekstrapolira iz temperature procesorja.

Informatika (in na splošno podatkovna pismenost) ima ključno vlogo pri razvoju trajnostno naravnane miselnosti in kompetentnosti. Še posebej je bistveno razumevanje podatkov in njihove verodostojnosti. Danes je to še toliko pomembnejše zaradi zanikanja podnebnih sprememb in obilja lažnih informacij.

Učitelj lahko zato dijake usmerja, da poiščejo podatke, povezane s trajnostnim razvojem, na relevantnih straneh, na primer na:

- » <https://www.copernicus.eu/en>,
- » <https://overshoot.footprintnetwork.org>.

Podatkovno modeliranje

Dijake seznanimo s sistemi za upravljanje podatkov. Dijake naučimo, kako zgraditi podatkovni model. Dijaki pri tem spoznajo:

- » modeliranje podatkovne baze,
- » izdelavo globalnega in konceptualnega modela (PB) podatkovne baze (ER model; entitete, atributi in relacije),
- » izdelavo logičnega modela (preslikava entitet v tabele, primarni in tuji ključi, podatkovni tipi atributov; normalizacija),
- » zbiranje in shranjevanje podatkov (fizični model: polnjenje pripravljenih tabel v podatkovni bazi),
- » prikazovanje in preoblikovanje ter sklepanje (poizvedbe SQL, indeksne tabele),
- » identifikacijo podatkovnih virov,
- » normalizacijo in zagotavljanje integritete podatkov,
- » vzdrževanje/ohranjanje podatkovnega modela.

Namig: Avtentični problem se lahko nanaša tudi na teme s področja trajnostnega razvoja ali drugih področij skupnih ciljev.

PRIKAZOVANJE IN PREOBLIKOVANJE 70

CILJI

Dijak:

○: obdeluje podatke in jih prikazuje na različne načine.

STANDARDI ZNANJA

Dijak:

- podatke obdela tako, da:

- » izloči odvečne podatke,
- » uporabi osnovne matematične operacije in funkcije (vsota, aritmetična sredina, min, max, count),
- » izračuna srednje vrednosti (mediana, modus),
- » podatke grafično in tabelarno prikaže,
- » zna združiti medsebojno odvisne podatke.

TERMINI

- grafični prikaz podatkov
- obdelava podatkov
- funkcija
- mediana
- modus
- vsota
- min
- max
- count
- aritmetična sredina

DIDAKTIČNA PRIPOROČILA ZA SKUPINO CILJEV

Zajemanje in shranjevanje podatkov je osnova za prikazovanje, transformacijo in interpretacijo podatkov.

S transformacijo podatkov:

- » razkrivamo in poudarjamo bistvene značilnosti podatkov,
- » olajšamo oziroma omogočimo nadaljnjo obdelavo z računalniki,
- » popravljamo in odstranjujemo morebitne napačne in nepopolne podatke.

Dijakom pokažemo, kako poiskati napačne meritve, na primer s preurejanjem tabele (opazimo odstopajoče številke v posamičnih stolpcih), s škatlami z brki (box plot), ki pokažejo izstopajoče primere ali, na primer, z razsevnim diagramom, ki pokaže odstopanja od trendov (npr. v prikazu, ki primerja starost in višino, vidimo osebe, ki so znatno prenizke ali previsoke za svojo starost).



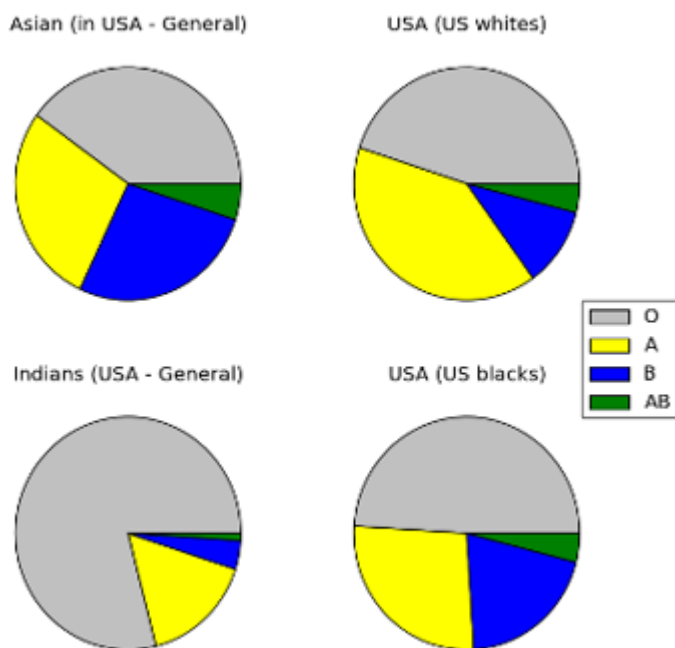
Pri izrazih, kot so mediana, aritmetična sredina in modus, niso bistvena njihova imena in formule, temveč pomen. Razložimo jim, na primer, razliko med aritmetično sredino (povprečjem) in mediano. Trditev »*več kot polovica plač je podpovprečnih*« je resnična, ker pomeni le, da je mediana nižja od aritmetične sredine. Na aritmetično sredino pomembno vpliva nekaj izjemno visokih plač, zato je mediana boljši kazalnik kot aritmetična sredina (povprečje).

Vizualizacij ne uporabljamo za okraševanje besedil, temveč nam omogočajo opaziti lastnosti podatkov, ki bi jih zgolj z opazovanjem tabel in števil težje odkrili. Primer slabe vizualizacije so krožni prikazi (tortni diagrami), ki so »praktično vedno manj informativni od običajne tabele s številkami« pravi Edward Tufte, strokovnjak za vizualizacijo podatkov. Spodnji prikaz bi bil brez števil neuporaben, predvsem pa številkam ničesar ne doda, zato je nesmiseln.

Sector Allocation of Holding		
FINANCIALS	21.45%	NON-CYCLICAL CONSUMER GOODS
CYCLICAL SERVICES	14.17%	INFORMATION TECHNOLOGY
RESOURCES	9.61%	GENERAL INDUSTRIAL
UTILITIES	3.83%	BASIC INDUSTRIES
NON-CYCLICAL SERVICES	3.67%	CYCLICAL CONSUMER GOODS

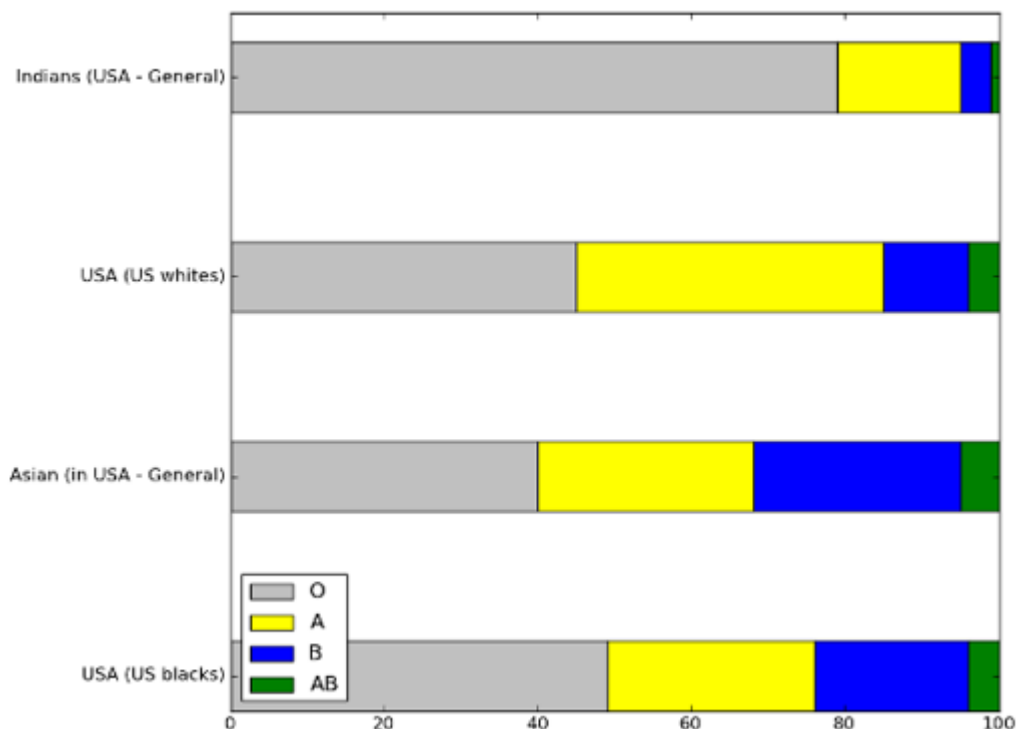
Krožni prikazi (tortni diagrami) so posebej neprimerni za primerjave distribucij, saj zahtevajo, da z očmi prehajamo od slike do slike, kot na primer na spodnjem prikazu deležev krvnih skupin med različnimi skupinami prebivalcev ZDA.





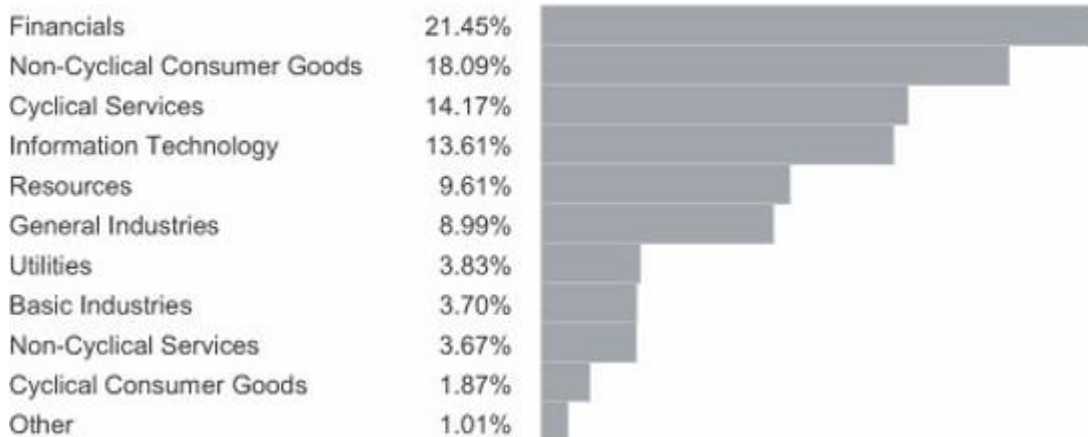
Primerne preproste vizualizacije so škatla z brki, histogram in razsevni prikazi (angl. *scatter plot*).

- » Škatla z brki kaže osnovne značilnosti (aritmetična sredina, mediana, standardni odklon ipd.). Ni pomembno, da si zapomnimo, kako je prikazan posamezni element (ali kažejo brki standardni odklon ali kvartile), saj za to ni »standarda«; pomembno je, da znamo, na primer, interpretirati vzporedno postavljene prikaze, ki kažejo, na primer, razlike pri določenem podatku za različne države.

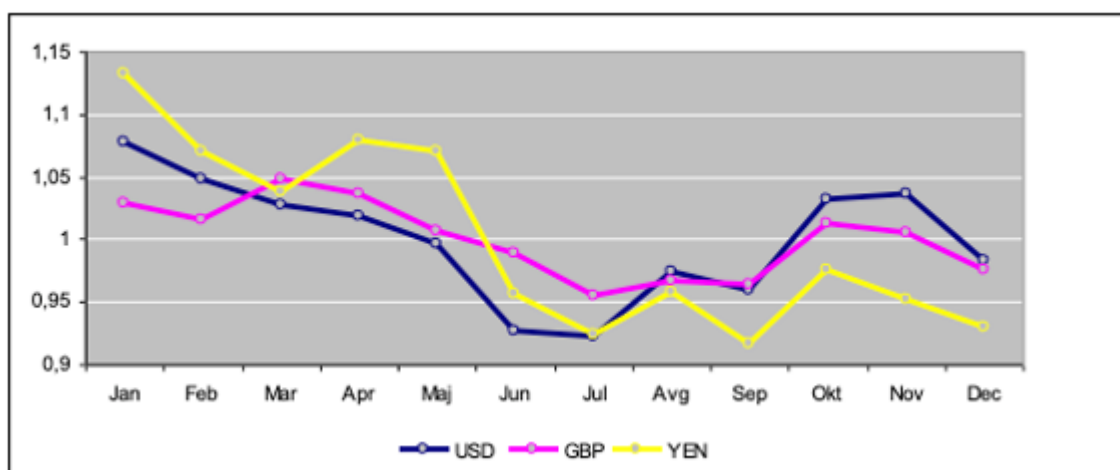


- » Histogrami (za številčne spremenljivke) so posplošitev škatle z brki, saj ne kažejo le statistik (srednje vrednosti, odkloni), temveč potek celotne distribucije. Pokažemo lahko primer simetrične (npr. Gaussove) distribucije, asimetrične distribucije in distribucije z dolgim repom; obstaja nekaj izjemno priljubljenih knjig in ogromno knjig z malo bralci.

Sector Allocation of Holding



- » Histogrami (za kategorične spremenljivke) omogočajo lažjo primerjavo. Podatke na zgornjih slikah lažje primerjamo s histogramoma.
- » Razsevni prikazi kažejo povezavo med dvema spremenljivkama, na primer povprečnim časom šolanja in pričakovano življenjsko dobo po različnih državah.
- » Črtno diagrame uporabljamo, kadar so podatki urejeni po zaporedni neodvisni spremenljivki, kot je čas — na primer za prikaz gibanja tečajnega razmerja med evrom in drugimi valutami skozi leto.



PRIKAZOVANJE IN PREOBLIKOVANJE 210

CILJI

Dijak:

○: prikazuje, transformira in interpretira podatke.

SC (4.1.2.1)

STANDARDI ZNANJA

Dijak:

- preoblikuje podatke s pomočjo:

- » agregacije (združevanja),
- » uporabe matematičnih operacij,
- » zamenjave ali odstranjevanja manjkajočih vrednosti,
- » prepoznavanja in odstranjevanja napačnih podatkov,
- » združevanja medsebojno odvisnih podatkov;

- prikaže podatke v ustrežnejši obliki in ob tem:

- » izbere ustrezne podmnožice podatkov (podatke filtrira),
- » podatke organizira in razvršča,
- » izračuna statistične vrednosti (standardni odklon, kvartili),
- » odkriva in razlaga vzorce v podatkih s pomočjo vizualizacije;

- na primerih pojasni, kako lahko preoblikovanje podatkov in vizualizacija vplivata na njihovo interpretacijo.

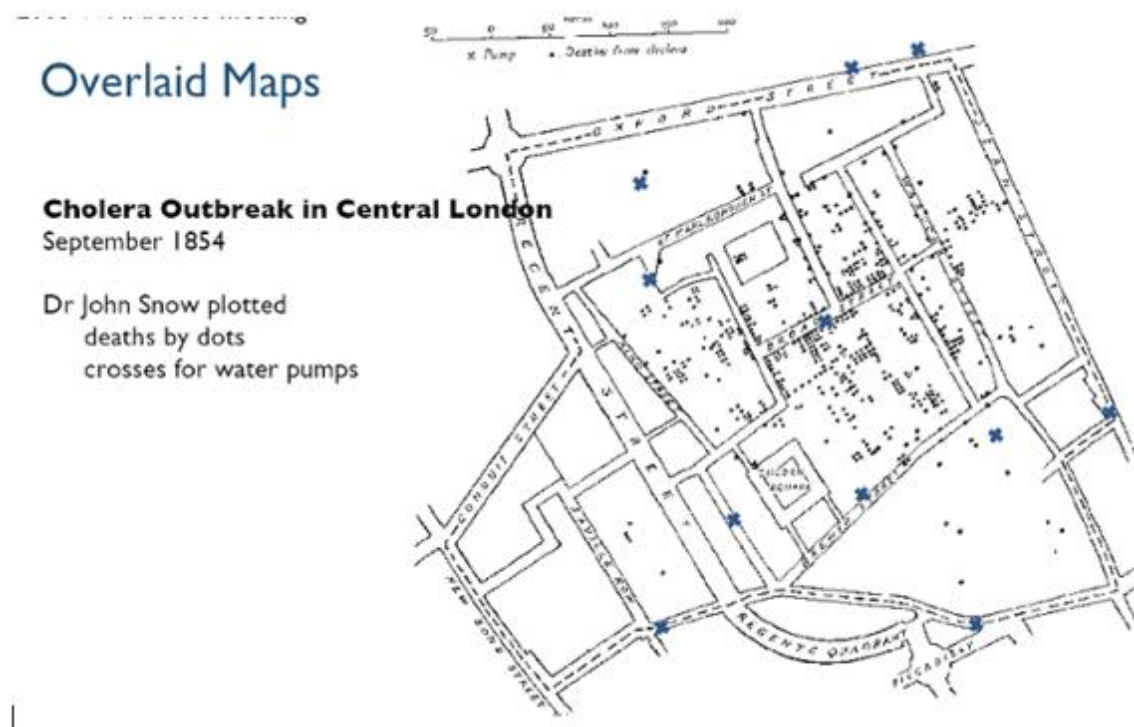
TERMINI

○ filtriranje podatkov ○ razvrščanje ○ vizualizacija ○ združevanje (agregacija) podatkov ○ strojno učenje

DIDAKTIČNA PRIPOROČILA ZA SKUPINO CILJEV

Vizualizacija podatkov na barvnem zemljevidu (choropleth): prikažemo lahko vremenske podatke (povprečna poletna temperatura v različnih državah), uvrstitve na zadnjem svetovnem prvenstvu v nogometu, lokacije večjih potresov (odkrivanje geografskih prelomnic).

Podatki na zemljevidu s točkami. Dr. John Snow je ob izbruhu kolere v Londonu leta 1854 na zemljevidu označil vse hiše, v katerih se je pojavila okužba, ter tako našel javni vodnjak, iz katerega je izvirala.



Elementi vizualizacije morajo slediti konvencijam in ne nalagati dodatnega kognitivnega bremena. Če barve točk na razsevnem diagramu predstavljajo ženske in moške, naj bodo te rožnate (ali rdeče) in modre, ne obratno. Kot zanimiv primer, kako težko je slediti slabo izbranim grafičnim elementom, se lahko dijaki preizkusijo v Stroopovem testu: povedati morajo, s kakšno barvo je napisana posamična beseda na spodnji sliki, torej »rdeča«, »rumena«, »črna«, »oranžna« in tako naprej. (Kar je napisano, morajo ignorirati.)

ZELENA	RJAVA	MODRA	VIJOLIČNA
RDEČA	ČRNA	ZELENA	MODRA
RUMENA	RJAVA	ČRNA	ORANŽNA
MODRA	VIJOLIČNA	RDEČA	ZELENA
ČRNA	MODRA	ZELENA	RJAVA
ČRNA	MODRA	ČRNA	MODRA
RUMENA	MODRA	RDEČA	VIJOLIČNA
RJAVA	OKER	MODRA	ORANŽNA

Opozorimo tudi na problematične (namerno ali nenamerno manipulativne) napake pri vizualizaciji, kot so:

- » Pretirano poudarjanje razlik zaradi prevelikega merila na skali.
 - » Primer prikaza tržnega deleža konkurenčnih borznih hiš na osnovi istih podatkov:
- » Spodnji prikaz pretirano poudarja spremembe.

[>](https://aplikacijaun.zrss.si/api/link/5osgqu8)
- » Spodnji prikaz ima korektno izbrano merilo (0 % - 100 %).

[>](https://aplikacijaun.zrss.si/api/link/zwl6knx)
- » Krivulji na spodnjem prikazu imata različni skali (54–60 oz. 30–34) in se navidezno sekata v letu 2001.

Desni graf primerja vsoto plač dvajsetih najbolje plačanih s plačo enega povprečnega delavca (poleg tega pa 850 000 / 35 500 ni 300, temveč 239 :).

[>](https://aplikacijaun.zrss.si/api/link/j857024)

Nekaj podrobneje razdelanih primerov analize podatkov z vizualizacijo:

- » košarkarji: <https://pumice.si/aktivnosti/kosarka>,
- » razvitost držav: <https://pumice.si/aktivnosti/razvitost-drzav>,
- » slovenski priimki: <https://pumice.si/aktivnosti/priimkoslovje/> (videi: <https://s.biolab.si/priimki-video>),
- » vremenski podatki: <https://pumice.si/aktivnosti/podnebnipasovi>.

Učitelj naj pripravi tudi dejavnosti s področja trajnostnega razvoja, kjer spodbuja sistemsko mišljenje ^{SC} (2.2.1.1) in razvoj kritičnega mišljenja ^{SC} (2.2.2.1) tako, da dijaki podatke analizirajo in jih interpretirajo.

SKLEPANJE IN MODELIRANJE 70

CILJI

Dijak:

I: oblikuje model podatkov, s pomočjo katerega išče odgovore na svoja vprašanja.

STANDARDI ZNANJA

Dijak:

- » z uporabo abstrakcije posploši opisno predstavljen problem, izdela model in argumentirano predstavi izbrano rešitev,
- » opiše ugotovitve, do katerih je prišel s pomočjo podatkov in modela.

TERMINI

- podatkovno modeliranje

DIDAKTIČNA PRIPOROČILA ZA SKUPINO CILJEV

Predlagamo, da se dijaki v prvem letniku seznanjajo s podatki in analizo s pomočjo preglednic (koncept tabele, ki vsebuje podatke, funkcije in grafikoni) in spletnih orodij za vizualizacijo.

Pri tem naj razvijajo preprost avtentični (podatkovni) model z uporabo preglednic in ročnim vnosom podatkov ali z uporabo obstoječih podatkov v preglednici.

Prikazovanje podatkov naj izvedejo z njihovim formatiranjem in prikazom v obliki grafikona. Podatke preoblikujejo z uporabo vgrajenih funkcij oz. izbranih orodij.

Namig: Avtentični problem se lahko nanaša tudi na teme s področja trajnostnega razvoja ali preostalih področij skupnih ciljev.

SKLEPANJE IN MODELIRANJE 210

CILJI

Dijak:

- : formalizira problem, ki ga namerava raziskovati, in določi ter zbere potrebne podatke,
- : na osnovi podatkov gradi modele za izdelovanje napovedi in prepoznavanje vzorcev v podatkih,
- : vrednoti uporabnost modela, ga razloži in izboljšuje.

STANDARDI ZNANJA

Dijak:

- določi potrebne podatke na osnovi opisno predstavljenega problema;
- zbere in shrani ustrezne podatke za pripravo modela;
- izdelava model za podan problem tako, da:
 - » podatke preoblikuje v ustrezno obliko,
 - » izbere primerno obliko modela,
 - » z ustreznim orodjem sestavi model;
- ovrednoti in uporabi zgrajeni model tako, da:
 - » določi kriterije uporabnosti modela,
 - » analizira model in oceni uporabnost,
 - » uporabi model za izdelavo napovedi.

TERMINI

- modeliranje realnosti ○ napovedovanje dogodkov ○ preverjanje teorij ○ kriterij uporabnosti modela
- analiza modela ○ omejitve modela ○ izboljševanje modela ○ gručenje ○ klasifikacija

DIDAKTIČNA PRIPOROČILA ZA SKUPINO CILJEV

Pri modeliranju običajno uporabljamo tabelarične podatke: primeri (vrstice tabele) so opisani z lastnostmi (stolpci). Če imamo poleg tega ciljno lastnost (izid), ki jo želimo napovedati iz podanih lastnosti, gre za »nadzorovano modeliranje«, če nas zanimajo le vzorci v podatkih, na primer sestavljanje gruč primerov, ki so si med seboj podobni, pa »nenadzorovano«.

Nadzorovano učenje

Za nadzorovano učenje uporabimo primerne avtentične podatke, na primer napovedovanje igralnega položaja košarkarja glede na njegove lastnosti (<https://pumice.si/aktivnosti/kosarka>), ali podatke, povezane z določenim predmetnim področjem. Za začetno predstavitev modelov so primerni tudi preprostejši sintetični primeri, na primer prepoznavanje različnih vrst štirikotnikov (<https://pumice.si/aktivnosti/stirikotniki>) ali živali (<https://pumice.si/aktivnosti/zivalski-kljuc>).

Kot prvi napovedni model je smiselno predstaviti klasifikacijsko drevo, saj je končni model intuitiven, poleg tega pa je lahko razumljiv tudi postopek njegove gradnje. Ob njem razložimo idejo strojnega učenja: računalnik sestavi formalni model na podlagi »učnih primerov«, potem pa lahko ta model uporabi za klasifikacijo novih primerov.

Smiselno je nadaljevati z linearnimi modeli, na primer z naivnim Bayesovim klasifikatorjem ali logistično regresijo, ki jih predstavimo s pomočjo nomogramov: linearni modeli napovedujejo izid tako, da za vsako lastnost dodelijo določeno število točk in primer pripada ciljnemu »razredu«, če število točk preseže določen prag. Sestavljanje modela (»učenje«) poteka tako, da računalnik na podlagi učnih primerov določi število točk, ki jih prinese posamična lastnost, in prag, potreben za ciljni razred.

Tretji model, ki ga lahko pokažemo, je razvrščanje na podlagi najbližjih sosedov: za vsak primer poiščemo vnaprej določeno število učnih primerov, ki so mu glede na opazovane lastnosti najbolj podobne. Primer pripada tistemu razredu, ki mu pripada večina sosedov.

Omenimo lahko naprednejše model, kot so naključni gozdovi, metode podpornih vektorjev in umetne nevronske mreže.

Prilagajanju hiperparametrov (npr. parametrov rezanja drevesa) se čim bolj izognemo do takrat, ko obravnavamo kriterije uspešnosti (glej spodaj).

Nenadzorovano učenje

Pokažemo predvsem hierarhično gručenje, za kar uporabimo podatke o razvitosti držav (<https://pumice.si/aktivnosti/razvitost-drzav> oz. <https://notes.biolab.si/books/ui-za-ucitelje>), o pogostosti priimkov po slovenskih občinah (<https://pumice.si/aktivnosti/priimkoslovje>) ali podnebne pasove (<https://pumice.si/aktivnosti/podnebni-pasovi>). Razložimo postopek gručenja in tudi analizo njegovih rezultatov (glej predvsem primere ob podatkih o razvitosti držav).

Koncept vložitve

Če podatki niso v tabelarni obliki, jih je treba vanjo spremeniti. Primer za to so slike in besedila. Ideja vložitve (embedding) je, da objekte opišemo z večjim številom značilk (lastnosti), ki jih dobimo s pomočjo globoke nevronske mreže. Ker je značilk preveč (tipično število za slike je okrog 2000), pomena posamičnih značilk ne moremo razumeti, vse skupaj pa predstavlja opis vsebine slike.

Za pouk pripravimo primerno zbirko slik. S primernim programom (npr. Orange) izračunamo njihovo vložitev, potem pa uporabljamo metode nadzorovanega ali nenadzorovanega učenja, ki smo jih spoznali pred tem. Primer že pripravljenih slik so slike tradicionalnih slovenskih hiš (<https://pumice.si/aktivnosti/vrste-his>).

Kriterij uspešnosti

Dijaki bodo kot kriterij uspešnosti predlagali delež pravih napovedi (klasifikacijsko točnost). Napeljemo jih na to, da niso vse napake enake: spregledati raka je hujša napaka kot diagnosticirati raka pri nekom, za katerega se izkaže, da ga nima. Uvedemo koncept matrike zmot (glej tudi <https://pumice.si/aktivnosti/kosarka>) ter – v primeru dveh kategorij, med katerima je ena ciljna, »pozitivna« – pravih in lažnih pozitivnih in negativnih primerov, iz katerih izpeljemo natančnost in priklic (precision in recall).

Naprednejše mere, kot sta F1 in AUC, so za srednjo šolo prezahtevne, lahko pa jih omenimo v dodatnem materialu.

Poudarimo, da modelov ne smemo preizkušati na primerih, na katerih smo jih gradili, saj s tem preferiramo modele, ki se pretirano prilagodijo podatkom: namesto da bi posploševali in iskali vzorce, si »zapomnijo« konkretne primere. Predstavimo idejo prečnega preverjanja (cross validation) ter ponovitev testov na več ločenih naključno izbranih učnih in testnih množicah.

Tu lahko pokažemo, kako spreminjanje parametrov gradnje modela (npr. parametrov rezanja drevesa) vpliva na njegovo točnost.





ALGORITMI IN PROGRAMIRANJE

OBVEZNO

OPIS TEME

Dijaki spoznavajo modeliranje avtentičnih problemov kot računalniško-informacijskih problemov, razvijajo in uporabljajo algoritme ter jih implementirajo v programskih jezikih. Proces reševanja problemov vključuje izdelavo ustreznega modela v računalniško-informacijskem svetu, izbiro in obdelavo podatkov, razdelitev kompleksnih problemov na manjše enote, kombiniranje obstoječih rešitev v rešitev novega problema in analizo različnih pristopov reševanja problemov. Pri tem razvijajo računalniško mišljenje.

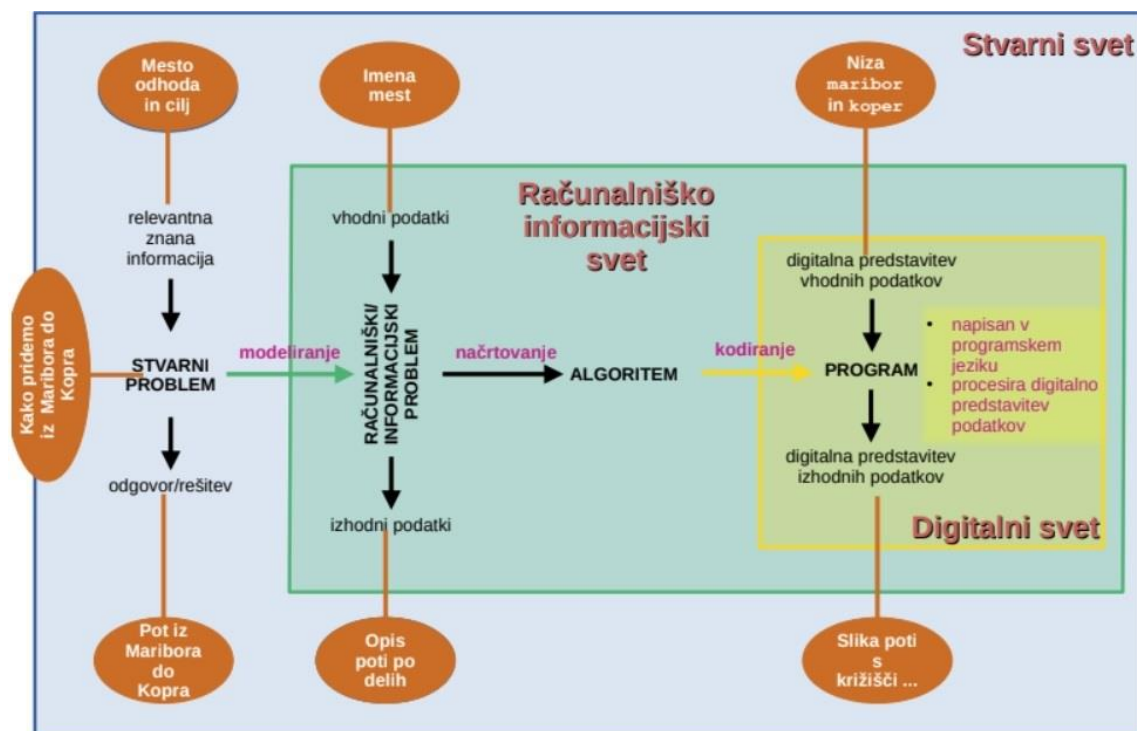
DODATNA POJASNILA ZA TEMO

Treba je poudariti, da tema Algoritmi in programiranje nikakor ni zasnovana kot zaporedno izvajanje posameznih skupin ciljev, temveč kot njihovo medsebojno prepletanje. Skupine ciljev so oblikovane izključno za opis učnih ciljev in standardov.

Izhodiščna (neformalna, a morda učinkovita) motivacija, ki jo lahko uporabimo za temo, je: računalnik je hiter, natančen, a »neumen«, človek pa je počasen, površen, a inteligenčen; s programiranjem kombiniramo zmožnosti računalnika in človeka ter tako lahko dobimo nekaj, kar je hitro, natančno in še inteligentno.

Pri celovitem pogledu na temo izhajamo iz prispevka Andreja Brodnika, Andrewa Csizmadie, Geralda Futscheka, Lidije Kralj, Violette Lonati, Petra Micheuza, Mattie Monge Programming for All: Understanding the Nature of Programs (<https://arxiv.org/abs/2111.04887>) in tam prikazane sheme »Od stvarnega problema do programa«.





DIDAKTIČNA PRIPOROČILA ZA TEMO

Poudarek pri programiranju je na razvoju računalniškega mišljenja in usvajanju osnovnih programerskih konceptov, kar se, ne glede na izbrano okolje in programski jezik, ne spreminja. Nikakor ni namen, da bi dijaki »postali programerji«, ampak jim s to temo želimo predstaviti osnovne koncepte in prijeme pri programiranju ter pri tem razvijati njihovo algoritmično mišljenje. Hkrati je cilj razvijanje računalniškega mišljenja in reševanja problemov tako z individualnim kot skupinskim, tj. sodelovalnim delom. Na področju programiranja in poznavanja ter razumevanja (razvijanja) programerskega procesa se tako spodbuja t. i. naravnost k rasti (npr. <https://aplikacijaun.zrss.si/api/link/lt3vota>).

Izbiri orodij (programskih jezikov, okolij, pristopov ...) je smiselno prilagoditi predznanju, zmožnostim in potrebam dijakov. Enako velja za izbrane zgledne. Tako v sklopu izvedbe učnih ur izbiramo med fizičnim računalništvom, uporabo robotov, programirljivim 3D-svetom in podobno. Uporabljamo lahko tako programiranje z delčki kot tekstovno programiranje. Učitelju, ki poučuje to temo, priporočamo, da se glede uporabljenih orodij in pristopov odloči za ustrezno kombinacijo in potem poučevanje prilagodi izbiri.

Pristopi za poučevanje programiranja

Učiteljem priporočamo, da preberejo članek (angl. *whitepaper*) Milesa Berryja, Paula Curzona, Quintina Cuttsa, Celie Hoyles, Simona Peytona Jonesa in Shahneile Saeed »*Practical programming in computing education*«, NCC Academic Board, februar 2022:

- » v razdelku 5, »Tips and principles available on the web«, je navedenih nekaj gradiv, ki naj bi jih poznal vsak učitelj, ki poučuje algoritme in programiranje – gradiva so na <https://aplikacijaun.zrss.si/api/link/3krfv6q> tudi na kratko predstavljena,

- » v razdelku 6 je spisek knjig, ki pokrivajo poučevanje računalništva in informatike nasploh – priporočamo, da omenjene knjige učitelji vsaj prelistajo.

Prav tako je smiselno kot literaturo za izvedbo te teme uporabiti:

- » *The Big Book of Computing Pedagogy*, Raspberry Pi Press, 2021, <https://aplikacijaun.zrss.si/api/link/sldcd5c>,
- » *The Big Book of Computing Content*, Raspberry Pi Press, 2022, <https://aplikacijaun.zrss.si/api/link/3ylax8y>.

Nekaj »prijemov«, ki jih priporočamo pri izvedbi pouka:

- » V teh priporočilih omemba orodij, s katerimi lahko dosežejo cilje posamezne teme, ni »navodilo«, naj izvajalci uporabijo ravno te; učitelji naj se odločijo na podlagi osebnih izkušenj in svojih zmožnosti (oz. šolskih tehničnih zmožnosti, tj. na podlagi opreme oz. orodij, ki jih imajo na razpolago).
- » Sledenje priporočilom metode **PRIMM** (Predict, Run, Investigate, Modify, Make): Naloge tipa »napiši program« pridejo na vrsto šele na koncu. Prej naj pripravijo naloge tipa »napovej, kaj bo naredil program«, »spremeni program« ... Izhajamo iz ugotovitve, da je učenje programskih jezikov podobno učenju tujih jezikov. In tam ne začnemo s pisanjem spisov, ampak z branjem, kombiniranjem besed, dopolnjevanjem stavkov, usvajanjem sintakse oz. pravil pisanja ... Podobna, a nekoliko poenostavljena je metoda UMC (Use-Modify-Create), predstavljena v članku Meghan Lund iz leta 2022 »Use-Modify-Create Framework« (<https://aplikacijaun.zrss.si/api/link/mikuc82>) in v prispevku D. Franklin idr. iz leta 2020 »An analysis of Use-Modify-Create pedagogical approach's success in balancing structure and student agency« v zborniku *Proceedings of the 2020 ACM Conference on International Computing Education Research*, str. 14–24 (<https://aplikacijaun.zrss.si/api/link/3ht9f18>).
- » **Parsonsovi problemi**: Osnovna ideja te oblike nalog je, da kodo razdelimo na koščke, ki jih je potem treba zložiti v pravem vrstnem redu.
Raziskave kažejo, da dosežemo isto raven poznavanja osnovnih konceptov (tako programskih jezikov kot algoritmov) kot pri »klasičnem« pouku programiranja, vendar hitreje.
Podpora za pripravo tovrstnih nalog je npr. orodje za sestavljanje učbenikov Runstone Academy (<https://runestone.academy/runestone>).
Več o tem si lahko preberete v/na:
 - » <https://aplikacijaun.zrss.si/api/link/lquhmcy>,
 - » B. Ericson in X. Hou: »Creating Parson's Problems with generative AI« (https://www.youtube.com/watch?v=ieFB_C2bq2Y),
 - » X. Hou, Z. Wu, X. Wang, B J. Ericson: »CodeTailor: Personalized Parsons Puzzles are Preferred Over AI-Generated Solutions to Support Learning«,
 - » B. J. Ericson, J. L. Pearce, S. H. Rodger, A. Csizmadia, R. Garcia, F. J. Gutierrez idr.: »Multi-Institutional Multi-National Studies of Parsons Problems«.

» **Programiranje v parih (angl. *pair programming*):** Eden narekuje, drugi tipka oz. sestavlja kodo. Nato po zaključeni nalogi oz. pretečenem času zamenjata vlogi. Ta pristop omogoča *razvijanje sodelovalnega, problemskega učenja in razvoj mehkih veščin 21. stoletja (komuniciranje, timsko delo)*.

» **Kako sošolec razvija kodo? Krožno programiranje:** Delamo v računalniški učilnici na različnih delih kode istega projekta na dogovorjen način ... Na znak učitelja zamenjamo delovno mesto (nadaljujemo delo na drugi kodi). S tem pokažemo več pomembnih načel pri programiranju, kot so *uvid v branje kode, dokumentiranje kode, pomembnost ustreznega poimenovanja spremenljivk, sledenja načrtu ...*

» **Agilni pristop** kot postopek razvoja programske opreme: Dijakom pokažemo, kako pri večjih programerskih projektih vsak dijak razvija svoj del programa – na začetku razdelimo problem na manjše podprobleme, ki skupaj tvorijo celoto oz. skupno rešitev

» Tipi programerskih nalog:

» kot smo omenili že pri metodah PRIMM in UMC, je smiselno, da dijakom zastavimo več nalog tipa:

- » kaj bo naredil ta program za določene podatke,
- » kaj je namen tega programa / te funkcije,
- » poišči primer podatkov, kjer program ne deluje,
- » popravi program,
- » spremeni program, da bo namesto tega počel to,
- » dopolni program;

» in manj tipa:

- » napiši program, ki ...

» **Enaki koncepti:** Pri praktičnem delu je dobro zastaviti več nalog, kjer algoritmično oz. konceptualno počnemo isto (gre za uporabo programskih vzorcev). Na primer:

» splošnejši pogled: pregled tabele, konkretne naloge pa so:

- i. poiščimo najdaljši niz v tabeli (seznamu),
- ii. poiščimo niz z največ samoglasniki,
- iii. poiščimo število z najmanjšo vsoto števk.

» **Algoritmične naloge in računalništvo brez računalnika:** Uporabimo čim več algoritmov, katerih postopek lahko razložimo brez računalnika.

» **Ustrezno poimenovanje spremenljivk:** Tako namesto koščka kode:

$x = 16$

raje uporabimo:

starost = 16

ali pa namesto:

if a < 10:

raje pišemo:

if ura < 10:

» **Konkretizacija primerov:** "Program za iskanje najboljše ocene v razredu" z realnimi podatki, kontekstom in praktično uporabo.

1. Namesto: "Algoritem za iskanje maksimalne vrednosti v seznamu"

Konkretizacija: "Program za iskanje najboljše ocene v razredu" z realnimi podatki, kontekstom in praktično uporabo.

2. Namesto abstraktnega:

python# Abstraktno

for i in range(n):

if seznam[i] > meja:

rezultat += 1

Konkretno:

python# Konkretno - štetje učencev, ki so dosegli več kot 60 točk na testu

for i in range(len(ocene)):

if ocene[i] > 60:

stevilo_pozitivnih += 1.

» **Dosledno oblikovanje pri pisanju kode:** V vseh gradivih, primerih ... se držimo priporočljivih in izbranih načinov oblikovanja (presledki okoli operatorjev, način komentiranja ...).

Programiranje v živo: Smiselno je, da ob aktivnostih, ki predvidevajo nastanek kode, koda nastaja v živo, med uro: pri tem se tudi namenoma motimo. Če potrebujemo večji del kode, ki ne potrebuje vmesne razlage, ta del kode pripravimo vnaprej in jo na način kopiraj in prilepi »natipkamo«.

Okolje (tolmač)

Priporočamo, da se izognete uporabi »standardnega« okolja IDLE (Integrated Development and Learning Environment) kot okolja za pisanje. Še posebej imajo težave z njim začetniki. Npr. Thonny (<https://thonny.org>)

(<https://aplikacijaun.zrss.si/api/link/cmcx924>) je začetniku veliko prijaznejše okolje. Seveda pa lahko uporabljate tudi spletna okolja. Teh je veliko (<https://www.online-python.com>, <https://onecompiler.com/python>, <https://aplikacijaun.zrss.si/api/link/0eljx74> so le trije primerki). Priporočamo pregled več spletnih okolij in izbiro tistega, ki najbolj ustreza vašemu načinu dela.

Koda

Tip znakov / pisava pri kodi: Kjer v gradivih navajamo kodo, jo pišemo z neproporcionalno pisavo (npr. Courier New, Consolas ...: <https://aplikacijaun.zrss.si/api/link/cjz1ulv>).

Znak za začetek/konec niza: Bolje je, da smo pri uporabi začetka/konca niza dosledni in vedno uporabljamo bodisi znak ' bodisi znak ". V vsakem primeru pa znotraj enega gradiva uporabljamo le en način (in ne razlagamo »nepomembnih« stvari, kot je znak ' znotraj znaka ").

Presledki okoli operatorjev: Bodimo dosledni pri uporabi. Lahko jih sploh ne pišemo ali pa jih pišemo povsod (priporočljivo). Načeloma priporočamo, da se držite pravil, kot jih določa Pylint (<https://pylint.readthedocs.io/en/stable>).

Torej ne pišemo (presledki okoli + so tu postavljeni tako, kot bi bili »naključni«:

```
print("Zmnožek "+ str(a)+" * " + str(15)+ " je enak " + str(c)+".")
```

ampak:

```
print('Zmnožek ' + str(a) + ' * ' + str(15) + ' je ' + str(c) + '.')
```

Še bolje je uporabiti f-niz:

```
print(f'Zmnožek {a} * 15 je enak {c}.')
```

Izpisovanje: Zaradi vse »telovadbe« s presledki, vejicami ... pri izpisovanju je najbolje, da enostavno takoj na začetku vpeljemo f-niz (v povezavi s print) in ga več ali manj uporabljamo. Torej namesto:

```
print("Zmnožek "+ str(a)+" * "+str(15)+ " je " + str(c)+".")
```

raje:

```
print(f'Zmnožek {a} * {15} je enak {c}.')
```

Import: Potreben je premislek o tem, katero od spodnjih dveh oblik uporabljati:

```
from random import *
```

```
import random
```

Morda je pri začetnih urah bolje, da pri uporabi drugih modulov uporabljamo:



```
from random import *
```

in potem:

```
kocka = randint(1, 6)
```

(seveda izključno kot recept). Tako dijakom ni treba uporabljati (in učitelju razlagati uporabe) pike (.) ter pisati `random.randint`.

Po drugi strani pa je prva oblika morda lažja za razlago (dva dela (`from in *` manj), če želimo uporabljati funkcije, ki niso del pythona, ampak neke programske knjižnice, npr. `random`); s pisanjem kode, kjer uporabimo piko (.), pa dijak lažje vidi, kam spadajo funkcije, ki jih uporablja.

» Skratka – potrebna sta premislek in preiščljena raba ene ali druge oblike (ali pa obeh) – odvisno pač od skupine učečih se.

Vsebina

Zgledi brez konteksta: Izogibamo se kodi brez konteksta. Bolje je, da se vidi določena uporabnost kode. Torej npr. namesto:

```
x = 5
```

```
y = 3
```

```
print(x + y)
```

raje:

```
levi_žep = 5
```

```
desni_žep = 3
```

```
print(f'Janez je v hlačah našel {levi_žep + desni_žep} frnikol.'
```

Uporabljeni primeri: Pogosto so v gradivih uporabljeni zgledi preveč »umetni«. Poskusimo uporabiti vsaj malo uporabne primere. Pri tem pa moramo biti pozorni, da pri reševanju ni težava v vedenju o konkretnem problemu (zato smo v zgledu v nadaljevanju fizikalno formulo eksplicitno navedli).

Recimo namesto »Preberi dve decimalni števili in izpiši njun produkt« za obravnavo istih konceptov raje uporabimo nalogo »Maja je bila popolnoma navdušena nad atrakcijo v adrenalinskem parku, imenovano Prosti pad, zato te prosi, da ji napišeš programček, ki bo izračunal in izpisal hitrost, ki jo dosežeš ob času t . Formula za izračun hitrosti je $v = g * t$, pri čemer je v hitrost, t čas (prebran podatek) in g gravitacijska konstanta ($g = 9,81$).«



Nekaj idej za zglede (gre izključno za primer postavljanja enostavnih problemov v kontekst) je pri Projektu Tomo na: https://www.projekt-tomo.si/problem_set/2813 (Mleko, Prosti pad, Razdelitve), https://www.projekt-tomo.si/problem_set/2814 (Hišnik, Potovanje), https://www.projekt-tomo.si/problem_set/2815 (Kosilo v šolski menzi, Pokojnina babice Tine, Grupa TNT) (govorimo o besedilih problemov, ne o konkretnih nalogah – ni nujno, da zahtevamo sestavljanje funkcij, kot je to predvideno pri omenjenih nalogah v Tomu).

Poimenovanje spremenljivk: Če ne uporabljamo »sintetičnih« zglédov, kot je:

```
a = 5
```

```
b = 15
```

```
c = a*b
```

```
print("Zmnožek "+ str(a)+" * " + str(15)+ " je enak " + str(c)+".")
```

ampak recimo:

```
a = 5
```

```
b = 15
```

```
plo = a*b
```

```
print(f'Ploščina pravokotnika {a} * {b} je {plo}.')
```

potem se lahko izognemo »neprimernim« imenom spremenljivk, kot so število a, b, c, d, e, x ...

Naključna števila: Dobro se obnese, če takoj na začetku pri obravnavi funkcij vpeljemo še:

```
import random
```

```
starost = random.randint(14, 19)
```

To naredimo brez razlage o psevdonaključnih številih, razlage o modulih. Povemo le, da imamo s tem na voljo funkcijo `random.randint(od, do)`, kjer kot rezultat dobimo naključno celo število med od in do – torej na istem nivoju, kot npr. razlagamo `math.sqrt()`.

Potem lahko delamo zanimive zglede.

Navajanje kode

Ni zaželeno, da kodo, ki je del gradiva, vstavljamo kot sliko. Bolje je, če je to tekst. Kot sliko jo navajamo le, če želimo poudariti, da morajo dijaki kodo pretipkati (morda koristno v določenih kontekstih).

Priporočena orodja in okolja



Za izvajanje programa priporočamo naslednja orodja oziroma okolja. Večina je razvitih v Sloveniji in so dostopna na naslovu <https://lusy.fri.uni-lj.si/ucbenik>.

Ime	Osnovni namen	Dostop	Opombe
Thonny	Okolje za pripravo in izvajanje programov v pythonu.	https://thonny.org	Začetniku prijaznejše okolje kot IDLE.
https://www.online-python.com/	Spletno okolje za pripravo in izvajanje programov v pythonu.	https://www.online-python.com/	Le kot primerek spletnega okolja. Obstaja jih veliko. Izberite sebi in svoji učni situaciji primerne.
https://onecompiler.com/python	Spletno okolje za pripravo in izvajanje programov v pythonu.	https://onecompiler.com/python	Le kot primerek spletnega okolja. Obstaja jih veliko. Izberite sebi in svoji učni situaciji primerne.
https://aplikacijaun.zrss.si/api/link/0elix74	Spletno okolje za pripravo in izvajanje programov v pythonu.	https://aplikacijaun.zrss.si/api/link/0elix74	Le kot primerek spletnega okolja. Obstaja jih veliko. Izberite sebi in svoji učni situaciji primerne.
Projekt TOMO	Zbirka različnih nalog v jeziku Python s samodejnim preverjanjem pravilnosti.	https://www.projekt-tomo.si/	
Putka	Zbirka srednje težkih in težjih nalog s tekmovanja ACM RTK in ACM UPM s samodejnim preverjanjem pravilnosti.	https://putka-rtk.acm.si/ https://putka-upm.acm.si/	
RTK	Zbirka lažjih in srednje težkih nalog s tekmovanja ACM RTK.	https://aplikacijaun.zrss.si/api/link/6uilib5	
Pišek	Zbirka nalog za programiranje z delčki s samodejnim preverjanjem pravilnosti.	https://pisek.acm.si/	
E-učbenik <i>Uvod v programiranje – blockly</i>	Učbenik za uvod v programiranje v jeziku Blockly.	https://aplikacijaun.zrss.si/api/link/vn7qbyd	
E-učbenik <i>Informatika 1</i>	Učbenik za uvod v programiranje (z jezikom python).	https://aplikacijaun.zrss.si/api/link/52ticdm#	Poglavje Programiranje in algoritmi.
Malina in Piton	Učbenik za uvod v programiranje (v jeziku Python) in uporabo fizičnega računalništva.	https://lusy.fri.uni-lj.si/ucbenik/rpi/	Vključuje sheme za izdelavo izdelkov za fizično računalništvo.
Runstone Academy	Orodje za pripravo spletnih e-učbenikov.	https://runestone.academy/	Veliko možnosti za pripravo različnih interaktivnih nalog s področja programiranja (Parsonsovi problemi, izvajanje kode, popravljanje, kratki odgovori ...).
Python Tutor	Orodje za spremljanje in vizualizacijo izvajanja kode.	https://pythontutor.com/	Podprti so tudi drugi jeziki!
Minecraft Education	Igra, v kateri je mogoče učiti programiranje.	https://education.minecraft.net/en-us	Omogoča učenje programiranja z delčki ali v jeziku python (prav tako tudi javascript). Omogoča vodeno učenje s pripravljenimi in vgrajenimi učnimi scenariji.

V nadaljevanju bomo opisali eno od možnosti poučevanja algoritmov in programiranja s pomočjo igre Minecraft Education.

Predlagamo, da temi Algoritmi in programiranje v prvem letniku namenite približno 24 ur. Igra Minecraft Education ima pri poučevanju programiranja kar nekaj močnih plati:

- » omogoča zelo nazorno demonstracijo in razlago programskih konceptov,
- » poenostavlja spremljanje izvajanja ukazov, saj lahko z opazovanjem agenta hitreje odkrivamo napake v programu,
- » programirati je mogoče tako z delčki kot v programskem jeziku Python,
- » omogoča skupinsko reševanje nalog in sodelovanje v skupnem svetu Minecrafta.

Uporaba igre Minecraft Education za učenje programiranja je preizkušen način, ki se je v praksi izkazal kot zelo primeren za dijake, ki s programiranjem še nimajo izkušenj.

Predlagamo, da čas, ki ga boste namenili za poučevanje programiranja, razdelite na dva dela.

V prvem delu (približno 12 ur) dijaki spoznajo s programskimi koncepti, v drugem delu pa naučeno znanje uporabijo pri projektu, s katerim poleg poglobljanja temeljnih znanj programiranja razvijajo tudi podjetnost (odkrivanje priložnosti, vrednotenje zamisli, motiviranost in vztrajnost, prevzemanje pobude, načrtovanje in upravljanje, izkustveno učenje, sodelovanje ...).

Prvi del: učenje posameznih konceptov programiranja

Predlagamo, da se dejavnosti izvajajo v blok urah.

Prvo uro se dijaki učijo oz. spoznajo izbran programski koncept s pomočjo vgrajenih vodičev/lekcij. Po zagonu Minecrafta vodiče/lekcije poiščemo tako, da kliknemo Lesson Library/Subject Kits/Computer Science, nato pa izberemo Block Coding, če želimo vaditi programiranje z delčki oziroma Coding with Python.

Vsak tak vodič ali lekcija praviloma zajema tri do štiri izzive, skozi katere dijaki spoznajo določeno snov oziroma programski koncept. V lekciji se dijaki praviloma ne učijo popolnoma samostojno, čeprav to lekcija omogoča. Lekcije namreč sledijo načelom problemskega učenja in lahko postanejo pretežke, če bi se takšnega učenja dijak lotil popolnoma sam.

Predlagamo, da vsako dejavnost učenja z vodiči vodi učitelj, ki sledi trem korakom:

- » Prvi izziv v vodiču je namenjen demonstraciji učitelja (angl. *Teacher-Led*; »How I Do«). V tem delu učitelj demonstrira reševanje prvega izziva, pokaže, kako si dijak lahko pomaga, če se mu zatakne, kako je treba brati navodila, kako se zapiše koda in kako se preveri pravilnost sestavljenega programa. Dijaki spremljajo njegovo delo in si po možnosti ustvarjajo zapiske.
- » Drugi izziv je namenjen vodenemu delu (angl. *Teacher Modeling*; »How We Do«). Učitelj demonstrira, dijaki delajo hkrati z učiteljem.
- » Tretji izziv je namenjen samostojnemu delu dijakov ob podpori učitelja. (angl. *Teacher Support*; »How You Do«). Dijaki se preizkusijo v samostojnem reševanju preostalih nalog, kjer učitelj podpira dijake le, če je to potrebno. Takšne lekcije dijaki običajno predelajo v eni šolski uri in se pri tem seznani s posameznim programerskim konceptom.

Smiselno je, da dijaki na koncu oddajo dokazilo o uspešno opravljeni lekciji. To je lahko zaslonski posnetek zadnje opravljene naloge, na kateri se vidi tudi agent z imenom dijaka.

Druga ura (v blok uri) je namenjena samostojnemu in občasno tudi skupinskemu preizkušanju naučenega v skupnem svetu igre Minecraft Education. Dijaki se prijavijo v skupni svet Minecrafta in po navodilih opravijo nalogo, ki jo pripravi učitelj. Učitelj pripravi takšen izziv, da morajo dijaki uporabiti pridobljeno znanje iz lekcije prejšnje ure, problem, ki ga rešuje dijak, pa je zanj nov. Učitelj kroži po razredu in po potrebi svetuje ter pomaga, hkrati pa spodbuja dijake k sodelovanju, medsebojni pomoči ter deljenju izkušenj. Ker je rezultat dejavnosti izdelek, lahko dijaki zaslonski posnetek izdelka in programa prav tako oddajo učitelju kot dokazilo o razumevanju posameznega koncepta in opravljeni nalogi.

Predlagamo, da se na opisan način izvede šest blok ur in s tem dijaki spoznajo ter utrdijo temeljne programske koncepte (zaporedje ukazov, zanke, pogojni stavki, spremenljivke, funkcije, itn. ...). Če ima učitelj na voljo manj ur, lahko po premisleku nekatere teme tudi izpusti. Dijaki tako spoznajo na primer le eno vrsto zank. S tem sicer predelajo manj različnih tem, a pridobijo minimalna potrebna znanja za projekt v drugem delu.

Drugi del: projektna naloga

Ko dijaki opravijo približno 12 ur dejavnosti, praviloma znajo dovolj, da lahko pričnejo samostojno graditi rešitve s programiranjem agenta v igri Minecraft Education. Najverjetneje čisto vsi dijaki po dvanajstih urah pouka še ne bodo resnično samostojno programirali. Zato pouk zastavimo tako, da dijake spodbujamo k sodelovanju in medsebojni pomoči. V drugem delu dijake seznanimo s problemom, ki ga bodo reševali tako samostojno kot tudi skupinsko. Ker želimo, da dijaki rešijo (sprogramirajo) precej zahteven problem, napovemo izdelavo skupinskega projekta. V tem delu učitelj predstavi projektni način reševanja problemov. Predstavi projektne korake in dokumentacijo, s pomočjo katerih bo reševanje problemov sistematično, hkrati pa bo lahko učitelj spremljal in po potrebi podpiral posamezne dijake. Celoten projekt izdelajo dijaki v šoli.

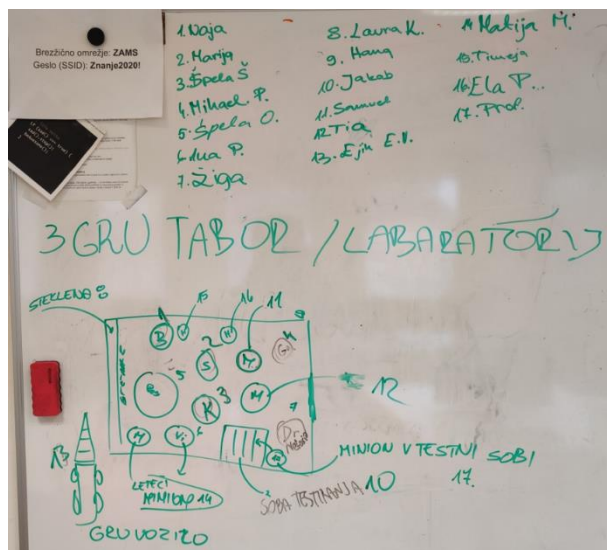
Prvi korak je **izbira problema**, ki se dogaja v skupini. Učitelj moderira proces izbire problema tako, da dijaki ne izberejo problema, ki ga zaradi časovne omejitve ali kompleksnosti ne bi bilo mogoče realizirati. Problem mora biti takšen, da ga lahko razdeli med dijake tako, da bodo morali sodelovati, hkrati pa se ne bodo čakali.



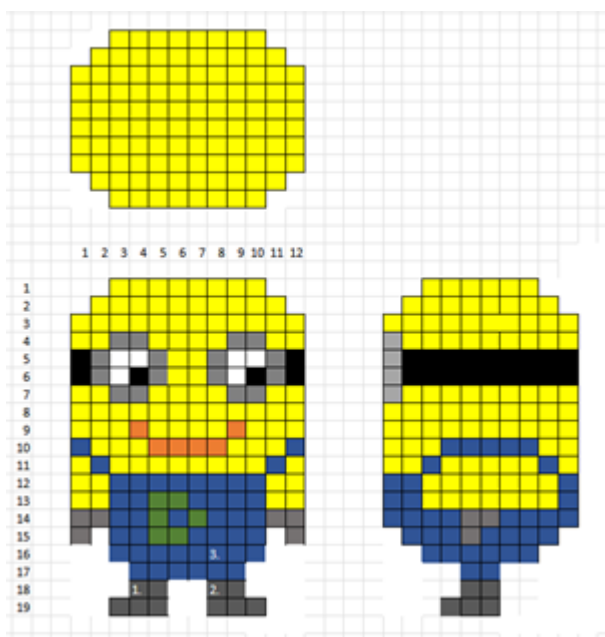
Najpomembnejši korak pri projektnem delu je **načrtovanje**. Dijakom je treba skozi proces izkustveno pokazati, da je načrtovanje tisto, ki zagotavlja izvedljivost kompleksnejših nalog. Zato skozi celoten proces izdelave projekta učitelj demonstrira posamezne korake. Tudi učitelj si izbere en problem (lahko mu ga tudi določijo

dijaki), nato pa vsak korak demonstrira, dijaki pa po njegovem zgledu nato opravijo svoj del naloge. Vsak dijak izdeluje svojo dokumentacijo projekta, v katero zapisuje svoj načrt in nato tudi dokumentira opravljeno delo. Tako ves čas nastajata dva vzporedna izdelka: praktični izdelek v Minecraftu in projektna dokumentacija v urejevalniku besedil. Ta dokument dijak naloži v oblak (npr. OneDrive v Microsoft 365) in povezavo odda učitelju v vpogled.

V procesu načrtovanja dijaki v skupini izbrano temo opišejo, saj morajo najprej dobro razumeti, kaj sploh je problem in kako ga bodo rešili. Nato sledi skupni načrt v obliki skice. Ob skici dijaki ozavešijo, katere elemente bo projekt vseboval, se dogovorijo, kako veliki bodo posamezni elementi ipd. Ob skici si razdelijo delo, pri čemer vsak prevzame vsaj približno enakovreden del naloge.



Sledi individualno načrtovanje, kjer vsak dijak samostojno podrobneje skicira svoj del naloge. Če je primerno, predlagamo izdelavo skice v Excelu, kjer je enostavno ustvariti »mrežo kvadratkov«, ki poenostavljajo gradnjo objekta iz kock v Minecraftu.



Ko je skica pripravljena, običajno ugotovimo, da je naloga prezahtevna, da bi jo sprogramirali v enem samem programu. Ponudi se odlična priložnost za osmislitev pomena dekompozicije. Vsak dijak svojo nalogo razdeli na manjše korake, kjer je posamezni korak dovolj enostaven, da ga je mogoče zapisati v obliki manjšega programa za agenta. Dekompozicijo seveda najprej demonstrira učitelj, nato pa tudi dijaki razbijejo problem na manjše, obvladljivejše probleme.

Ko dijaki v »projektni dokumentaciji« zapišejo vse korake, načrtovanje končajo in prične se naslednja faza – **izvedba**.

Učitelj predstavi »projektni cikel«, kjer iz načrta projekta izbere prvi korak in ga rešuje tako dolgo, dokler ga zadovoljivo ne opravi. Nato rešitev dokumentira in nadaljuje z naslednjim korakom iz projekta. Učitelj običajno pokaže reševanje enega ali dveh korakov. Dijaki tako spoznajo pričakovani način dela in prično z reševanjem, dokler ne opravijo vseh nalog. Dijaki celoten projekt opravijo v 5–6 srečanjih (blok urah). Dijaki opravljeno delo sproti dokumentirajo, kar zajema:

- » sestavljanje fotozgodbe (iz zaslonskih posnetkov) nastajanja izdelka po delovnih korakih in
- » oddajanje vseh programov, s katerimi je dijak rešil posamezne probleme/korake.

Ker dijaki dejavnosti izvajajo v skupnem svetu Minecraft Education, so soodvisni in lahko opazujejo druge dijake (in učitelja) pri reševanju problemov. Prav tako lahko tudi učitelj v živo spremlja napredovanje skupine in posameznikov, jim podaja povratne informacije in dijake usmerja k rešitvam.

Zadnji korak v projektu je **vrednotenje** opravljenega dela, kjer dijaki po podanih kriterijih pregledajo, kako jim je uspelo doseči zastavljene cilje.

S takim načinom dela dijaki razvijajo tudi druge spretnosti in digitalno kompetentnost. Na začetku prevladuje vadba osnovne koordinacije miška-tipkovnica ob premikanju avatarja in grajenju v 3D-svetu igre. Izkaže se, da nekateri dijaki to koordinacijo šele razvijajo. Dijaki se urijo tudi v uporabi spletne učilnice, saj v njej najdejo

navodila za delo in vanjo tudi oddajajo dokazila o učenju. Pri pripravi dokazil o učenju se urijo v zajemanju zaslonskih slik, obrezovanju, shranjevanju in oddajanju. Med projektno nalogo se učijo uporabljati dokumentni sistem v oblaku (npr. OneDrive v Microsoft 365), kjer ustvarjajo dokumente (besedilni dokument za dokumentacijo, preglednico za skico ipd.). Te spletne dokumente nato po navodilih dopolnijo z vsebino in oblikujejo (urejanje besedila s slogi ...), dokumente dajejo v skupno rabo ipd. Za razvijanje teh veščin je potreben čas in projektno delo pri informatiki je primerno za doseganje teh ciljev.

Opisani model učenja programiranja v igri Minecraft Education je preizkušen model, rezultati in izkušnje dijakov so zelo spodbudne.

PRIPOROČENI OZ. MOGOČI NAČINI IZKAZOVANJA ZNANJA (velja za celotno temo Algoritmi in programiranje)

Tako pri formativnem kot sumativnem preverjanju znanja je poseben poudarek na izdelavi takih artefaktov, kjer je mogoča izvedba z avtomatičnim agentom. Konkretno to pomeni, da naj bi dijaki bili sposobni ustvariti npr. program v programskem jeziku Python, s katerim bi demonstrirali svoje znanje in razumevanje koncepta zanke, ali pa pokazati svoje poznavanje modularnosti tako, da rešijo problem, ki predvideva izdelavo funkcije in klic le-te. Učitelj se lahko odloči tudi za igro Minecraft Education, kjer lahko dijaki programirajo agenta, ta pa ob zagonu programa v 3D-svetu izvaja ukaze. Spremljanje izvajanja ukazov in iskanje napak v Minecraft Education sta zaradi nazornosti postavljanja ali brisanja kock v 3D-svetu zelo poenostavljena.

V luči izzivov, kot so ChatGPT in druga generativna orodja, ki že sama izdelajo del programske kode je potrebno znova premisliti in prilagoditi preverjanje znanja. Kot ena izmed možnosti preverjanja znanja se ponujajo projektne naloge. Te od dijaka zahtevajo, da dokaže razumevanje in poznavanje programerskega procesa in priprave programske kode na višji ravni, kot jih zmorejo generativna orodja, sploh če poizvedbe niso pravilno zastavljene. Po eni strani ta orodja lahko programersko delo olajšajo in pospešijo, programiranje postane hitrejše in učinkovitejše, še vedno pa moramo od dijaka zahtevati razumevanje procesa in ustreznih konceptov.

Pri uporabi preverjanja znanja z domačim delom (domače naloge, projekti), ki ga načeloma priporočamo kot osnovni način tako formativnega kot sumativnega preverjanja znanja, se je treba zavedati, da imajo dijaki dostop do orodij generativne umetne inteligence. Računati na to, da bodo orodja dala napačne rezultate in bomo pri tem prepoznali nepravilno uporabo, pri nivoju problemov, ki jih zastavljamo dijakom, ni smiselno. Prav tako menimo, da, še posebej pri domačem delu, prepoved uporabe nima smisla. Priporočamo pristop, ko morajo dijaki svoje domače izdelke zagovarjati, torej ustno razložiti posamezne dele, prijeme ... Ta pristop, torej pojasnjevanje svoje kode, svojega izdelka, izbire algoritma, je že sam po sebi odličen način preverjanja znanja.

Pri ustnem preverjanju domačega dela priporočamo naslednje tipe nalog:

- » sledenje kodi (dijak pokaže, kaj njihova koda dela na nekem naboru podatkov),
- » iskanje napake v kodi (rešitev, podobna njegovi, a ima napako),
- » primerjanje svoje rešitve z drugo rešitvijo (z nekoliko drugačnim pristopom),
- » reševanje izomorfni problemov (dijaku zastavimo problem, katerega rešitev lahko dobimo z majhnimi spremembami njegove kode /za prvotni problem/).

ALGORITMI 70

CILJI

Dijak:

- : napoveduje rezultate/učinke danega algoritma,
- : sledi izvajanju danega algoritma,
- : analizira namen danega algoritma,
- : spreminja dani algoritem tako, da doda novo funkcionalnost in/ali reši problem z drugim naborom ukazov,
- : ustvarja algoritme, ki rešujejo dane probleme.

STANDARDI ZNANJA

Dijak:

- sledi izvajanju algoritma pri konkretnih podatkih in napove splošni namen algoritma;
- za podan problem ustvari (ali spremeni obstoječi) algoritem in pri tem:
 - » problem razloži in definira vhodne podatke, ki ga opredeljujejo,
 - » ustvari algoritem, ki rešuje dani problem,
 - » analizira pravilnost ustvarjenega algoritma,
 - » v algoritmu prepozna osnovne programske/algoritemske strukture: zaporedje ukazov, vejitev, ponavljanje.

TERMINI

◦ algoritem ◦ podatki ◦ algoritemske strukture ◦ podproblemi ◦ psevdokoda

DIDAKTIČNA PRIPOROČILA ZA SKUPINO CILJEV

Upoštevamo predvsem smiselni del splošnih priporočil za temo Algoritmi in programiranje.

Algoritme zapišemo na različne načine, kot so preprosto besedilo v smislu razlage v naravnem jeziku, z diagramom poteka, psevdokodo, zapis v obliki delčkov ...

Vsakdanji postopki so lahko podani neformalno in dvoumno; postopki, namenjeni računalniku, morajo biti podani formalno in nedvoumno (tako, da je vsako navodilo mogoče razumeti samo na en način). Programski jeziki so v nasprotju z naravnimi zastavljeni tako, da dvoumnosti niso mogoče.

Primeri: Vsakdanji postopki (npr. pomivanje posode, oblačenje, pot do šole ipd.), iskanje določene besede na neurejenem in urejenem seznamu besed, urejanje listkov z besedami na različne načine in z različno urejenostjo (dolžina besede, abecedni vrstni red)... Pri vsakdanjih postopkih je treba biti pozoren na izvedljivost, določenost, nedvoumen pomen ukaza ... Verjetno neustrezen (odvisno od načina predstavitve), a pogosto uporabljen zgled za algoritem, je kuharski recept. A tam imamo veliko nedoločenih operacij, npr. »po potrebi primešamo moko«, »po okusu dosolimo« ... Boljši zgled so npr. navodila za sestavljanje pohištva (ali modela iz legokock). Dijaki drug drugemu sestavijo navodila za premikanje, navodila za risanje ... Pri tem se lahko uporabi primerjava vsakodnevnih »problemov« z deljenjem na manjše »podprobleme«, recimo jutranja priprava na dan v šoli se lahko »razbije« na manjše probleme – vstajanje, oblačenje, pot do šole, enako se lahko naredi za dan v šoli (urnik je sestavljen iz več ur, posamezna učna ura iz drugih manjših enot itd.). Ozaveščanje dijakov na ustrezen način o tem, da so algoritmi prisotni v vsakdanjem življenju in da jih lahko delimo na manjše dele, ki prispevajo k »večjemu« cilju.

Pri razlagi je smiselna uporaba pristopov računalništva brez računalnika. Dobro se obnesejo dejavnosti z razvrščanjem skupine elementov po teži (primerjava različno težkih elementov).

Pri razlagi tematike je treba paziti na razliko med postopkom in algoritmom (izvedljivost, ustavljenost ...). Zato se izogibamo formalnim definicijam algoritma, poudarek naj bo na razumevanju (čeprav morda to pomeni »ohlapnejše« definiranje pojmov).



ALGORITMI 210

CILJI

Dijak:

- : za dani problem išče različne algoritme in primerja čas njihovega izvajanja,
- : kompleksnejše probleme deli na manjše podprobleme,
- : napoveduje obnašanje algoritma v odvisnosti od vhodnih podatkov,
- : analizira pravilnost delovanja algoritma in z različnimi postopki odpravlja napake.

STANDARDI ZNANJA

Dijak:

- za podani problem:

- » razloži bistvo problema in definira vhodne podatke,
- » opiše obliko pričakovane rešitve (izhodni podatki),
- » ustvari algoritem, ki reši dani problem, in pri tem kompleksnejše probleme razdeli na manjše podprobleme;

- opiše primere problemov, kjer obstaja več algoritmov za reševanje, in primerja čas njihovega izvajanja;

- utemelji pravilnost ustvarjenega algoritma.

TERMINI

○ algoritem ○ podatki ○ algoritemske strukture ○ podproblemi ○ psevdokoda

DIDAKTIČNA PRIPOROČILA ZA SKUPINO CILJEV

Upošteveno predvsem smiselni del splošnih priporočil za temo Algoritmi in programiranje.

Dijaki spoznavajo določene pristope pri razvoju algoritmov: rekurzijo, deli in vladaj, sestopanje, bisekcija, hitro razvrščanje (quick sort), urejanje z zlivanjem ... šele na koncu teme Algoritmi in programiranje. Ta cilj je izbirni. Dijaki namreč potrebujejo nekaj izkušenj s programiranjem in reševanjem problemov, preden je seznanjanje s temi pristopi smiselno.



SPREMENLJIVKE 70

CILJI

Dijak:

○: analizira vlogo spremenljivk in jih uporablja v programih.

STANDARDI ZNANJA

Dijak:

- uporabi spremenljivke tako da:

- » ustvari spremenljivko z ustreznim imenom, ki hrani vrednost podatka,
- » določi ustrezni tip spremenljivke (*celo število, realno število, niz znakov*),
- » upravlja spremenljivko (*priredi in spremeni vrednost, prebere vrednost, uporabi jo v ustreznih (aritmetično-logičnih) operacijah* oz. v poljubnih izrazih),
- » razlikuje med izrazom in vrednostjo izraza.

TERMINI

◦ spremenljivka ◦ prireditveni stavek ◦ tip spremenljivke ◦ podatkovni tip ◦ aritmetične operacije ◦ logične operacije

DIDAKTIČNA PRIPOROČILA ZA SKUPINO CILJEV

Upoštevamo predvsem smiselni del splošnih priporočil za temo Algoritmi in programiranje.

Treba je jasno povedati, da spremenljivka ohrani svojo vrednost, dokler ji vrednosti izrecno ne spremenimo. Na primer po zaporedju ukazov $a = 5$, $b = a + 1$, $a = 10$ ima spremenljivka b še vedno vrednost 6, ne 11. Pri izvajanju programa računalnik v spremenljivko shrani zgolj vrednost, originalni izraz pa se izgubi. Dijaki, ki so kdaj uporabljali elektronske preglednice, imajo namreč drugačne izkušnje pa tudi v matematiki je pojem spremenljivke drugačen kot pri programiranju.

Dobro se izkaže prvi del metode **PRIMM (Predict, Run, Investigate, Modify, Make)** – dan je enostaven program, dijaki napovejo njegov izid. Predavatelj nazorno (grafično, tabelarno) prikaže dogajanje v programu (ob prirejanju spremenljivk). Uporabno orodje za vizualizacijo spreminjanja spremenljivk je <https://pythontutor.com>.

Programi postanejo bistveno zanimivejši, ko vanje vpletemo še uporabnika. V nekaterih programskih jezikih (npr. python) se pri obdelavi vhoda srečamo s pretvorbo tipa (npr. niz -> celo število). Primeri: Program, ki

prebere ime in izpiše pozdrav s tem imenom; program, ki prebere leto rojstva in izpiše starost v letošnjem letu; program, ki prebere število prevoženih kilometrov ter začetni in končni čas (ure in minute), izpiše pa povprečno hitrost v km/h.

Prav tako že zgodaj uvedemo uporabo knjižnic pripravljenih metod, ki jih obravnavamo kot »črne škatle« (npr. `random`, `math`), in s tem še popestrimo zglede.

Probleme vedno konkretiziramo. Na primer:

- » program, ki med seboj zamenja vrednosti dveh spremenljivk (analogija z dvema polnima kozarcema). Pripravimo konkretno besedilo, ne »sintetičnega zgleда« – npr. *»Ko je Janezek napisal višina = in širina =, je ugotovil, da se je zmotil in da bi moral vrednosti spremenljivk višina in širina zamenjati. Kako naj to stori s programskimi ukazi?«*;
- » program za izračun srednje vrednosti treh celih števil (tukaj pride do izraza razlika med celoštevilskim in realnoštevilskim tipom) – a spet moramo problem »konkretizirati« (npr. Janez, Miha in Ana so tekmovali v metu krogle: 15 m, 18 m, 16 m. Koliko je povprečna razdalja?).

SPREMENLJIVKE 210

CILJI

Dijak:

- : prepozna probleme, ki zahtevajo hranjenje podatkov v tabelah in izvaja operacije nad njimi,
- : uporablja sestavljene podatkovne tipe (tabela),
- : uporablja osnovne algoritme, kot je npr. iskanje podatkov v tabeli (seznamu).

STANDARDI ZNANJA

Dijak:

- » **navede primer problema, ki zahteva hranjenje podatka v tabelah**, in demonstrira njegovo rešitev,
- » razloži nekaj tipičnih algoritmov za delo nad podatki v tabeli (iskanje podatkov z določeno lastnostjo, *urejanje*, **izpisovanje**).

TERMINI

◦ spremenljivka ◦ prireditveni stavek ◦ tip spremenljivke ◦ podatkovni tip ◦ aritmetične operacije ◦ logične operacije

DIDAKTIČNA PRIPOROČILA ZA SKUPINO CILJEV

Upoštevamo predvsem smiselni del splošnih priporočil za temo Algoritmi in programiranje.



KONTROLNE STRUKTURE 70

CILJI

Dijak:

- : uporablja kontrolne strukture, da doseže kompleksnejše obnašanje programa,
- : raziskuje primere problemov, kjer ne ve vnaprej, koliko ponovitev bloka ukazov bo potrebnih.

STANDARDI ZNANJA

Dijak:

- » izpiše določeno besedilo ali število,
- » ustvari program, ki se različno obnaša glede na vneseno vrednost podatka (npr. uporabi vejitev: če je podatek večji ali manjši od vrednosti 0, potem ...),
- » zapiše program, ki se zna sprehoditi po vseh podatkih in nad njimi izvede določeno operacijo (uporabi ustrezno vrsto zanke za npr. preštevavanje podatkov z določeno lastnostjo).

TERMINI

- zanka
- pogojni stavek
- števna zanka
- logični pogoji
- gnezdene strukture
- veriženje pogojev

DIDAKTIČNA PRIPOROČILA ZA SKUPINO CILJEV

Upoštevamo predvsem smiselni del splošnih priporočil za temo Algoritmi in programiranje.

Pogojne stavke je smiselno razdeliti na tri dele: if stavek; if-else stavek; veriga if-elif-else). Pri tem v zahtevani cilj spada le osnovna oblika (if), ostale oblike so izbirne.

Zanko while lahko razumemo bodisi kot ponavljajoč se pogojni stavek (diagrama poteka sta skoraj enaka) bodisi kot posplošitev števnih zank, za primere, kjer število ponovitev ni vnaprej znano. **Primeri:** simulacija nakupovanja do porabe razpoložljivega denarja; ugibanje števil; ...

Pogoj pri pogojnem stavku mora biti True ali False (izogibamo se implicitno določenim pogojem, kot so uporaba izraza z vrednostjo 0 kot "sinonim" za False, ali nepraznega seznama kot sinonim za True). Zato ob uporabi pogojnih stavkov ali zanke while hkrati uvedemo še logični tip, logične vrednosti in logične izraze.

Po izkušnjah dijaki s preprostimi števnimi zankami nimajo težav, se pa stvari zelo spremenijo, ko uvedemo gnezdenje zank in pogojnih stavkov (npr. stavek if v telesu zanke for), še bolj pa pri gnezdenju zank (for znotraj telesa for).

Uporaba veriženja pogojev, gnezdenje struktur itd. je stvar odločitve učitelja glede na skupino dijakov in ne spada med osnovne cilje.

Uporabni primeri: izračun (približka) števila π z izbiranjem točk v kvadratu; računanje vsote števil na vhodu; iskanje največjega ali najmanjšega števila; statistična verjetnost (met kovanca, met kocke ploščina ...).

Pri praktičnem delu je dobro zastaviti več nalog, kjer je sama struktura zanke enaka, v telesu pa počnemo različne stvari (gre za uporabo programskih vzorcev). Na primer:

- » preštejemo, koliko nizov ima le en samoglasnik (Ana ima seznam imen svojih sošolcev: ['Maja', 'Peter', 'Ines', 'Rok']). Koliko imen ima točno en samoglasnik?),
- » preštejemo, koliko števil v tabeli je pozitivnih,
- » poiščemo največje število (Učitelj je ocenil teste iz matematike: [95, 61, 10, 73, 92, 45, 12, 56]. Katera po vrsti je največja vrednost?; V trgovini imajo cene izdelkov: [2.50, 1.80, 5.20, 3.10]. Kateri izdelek je najdražji?),
- » poiščemo najdaljši niz,
- » ...

KONTROLNE STRUKTURE 210

CILJI

Dijak:

- : rešuje probleme, ki zahtevajo obdelavo podatkov v tabeli in pri tem razloži uporabo kontrolnih struktur,
- : primerja števno zanko in zanko, kjer število ponovitev v splošnem ni znano.

STANDARDI ZNANJA

Dijak:

- » v programskem jeziku zapiše večkratno opravljanje enakega opravila z uporabo zanke,
- » v programskem jeziku zapiše pogojno in večkratno izvajanje opravila,
- » v programskem jeziku zapiše algoritem, ki zahteva obdelavo podatkov v tabeli in rešitev razloži.

DIDAKTIČNA PRIPOROČILA ZA SKUPINO CILJEV

V didaktični literaturi ni enotnega soglasja o tem, ali naj se števne zanke (*for*) uvajajo pred zankami s pogojem (*while*). Vendar pa številni viri poudarjajo, da je vrstni red uvajanja teh konceptov odvisen od didaktičnih ciljev, predznanja učencev in konteksta poučevanja.

For zanke so primerne, kadar je število ponovitev znano vnaprej, medtem ko so *while* zanke bolj primerne za situacije, kjer število ponovitev ni določeno vnaprej.

MODULARNOST 70

CILJI

Dijak:

- : v programih uporablja obstoječe funkcije, ki jim poda ustrezne podatke kot parametre,
- : analizira potrebo po funkciji, ki izvede določeno opravilo,
- : sprogramira enostavno funkcijo in jo uporabi v programu.

STANDARDI ZNANJA

Dijak:

- **v programu uporabi klic obstoječe funkcije in pri tem poda ustrezne vrednosti parametrov;**
- na osnovi konkretnega problema analizira potrebo po funkciji/-ah in opiše rezultat delovanja posamezne funkcije tako, da:
 - » zapiše namen funkcije,
 - » navede parametre, ki so za to funkcijo potrebni;
- na osnovi danega opisa funkcije in njenih parametrov zapiše funkcijo in jo uporabi v rešitvi.

TERMINI

○ funkcija ○ klic funkcije ○ knjižnice funkcij ○ parametri ○ modul

DIDAKTIČNA PRIPOROČILA ZA SKUPINO CILJEV

Upošteevamo predvsem smiselni del splošnih priporočil za temo Algoritmi in programiranje.

Priporočamo zgodnjo vpeljavo funkcij (lahko še pred vsemi kontrolnimi strukturami), da lahko začnemo z enostavnimi primeri in je dovolj časa za utrjevanje tega zelo pomembnega pojma. Začnemo z opisom pomena že prej uporabljenih funkcij (input, int, str ...) in funkcij, ki jih »pridobimo« iz knjižnic (npr. random.randint(), math.sqrt()).

Po izkušnjah so funkcije za dijake precej abstrakten (in zato vsaj na začetku zahteven) koncept, zato ga je smiselno razbiti na tri dele:

- » funkcije brez parametrov in brez vračanja (motivacija za uvedbo funkcije je tukaj lahko enostavno takšna: imamo kos kode, ki se večkrat ponovi; da se temu izognemo, nam programski jezik omogoča, da kos kode poimenujemo in večkrat pokličemo);



- » funkcije s parametri in brez vračanja rezultata (tukaj je treba poudariti, da pri klicu funkcije parametri dobijo, kot svojo začetno vrednost, argument in da sprememba parametra ne vpliva na pripadajoči argument. Zato je dobro pokazati vsaj kak zgled, kjer za argument ob klicu uporabimo sestavljeni izraz (in je »očitno«, da povezave med argumentom in parametrom ne more biti). Npr. izračun diagonale pravokotnika z uporabo matematične funkcije `sqrt`. Pokažemo, da je koda:

```
c = math.sqrt(a*a + b*b)
```

načeloma enaka kodi:

```
vso = a*a + b*b
c = math.sqrt(vso)
```

- » funkcije z vračanjem vrednosti (marsikdo med dijaki ima težave z ločevanjem med izpisom vrednosti in vračanjem vrednosti; pogosto je s tem povezano tudi nerazumevanje razlike med spremenljivko in vrednostjo – vrnemo vrednost, ne spremenljivke!).

Primeri: Program, ki na več različnih mestih izpiše ime, priimek in poštni naslov (funkcija brez parametrov in brez vračanja); program, ki enako stori za različne osebe (funkcija s parametri, a brez vračanja); program, ki prebere koordinate oglišč pravokotnika in izračuna skupno dolžino vseh povezav med vsemi pari oglišč (funkcija s parametri in vračanjem).

Predvsem naj bi bil pri obravnavi koncepta funkcij poudarek na vlogi funkcij pri delitvi na podprobleme: demonstriramo (s »programiranjem v živo«) prednosti postopka, ko na določenem mestu potrebujemo rezultat, in za pridobitev slednjega le pokličemo (neobstoječo) funkcijo in nadaljujemo z razvojem. Temu sledi kasnejši razvoj te funkcije.

Zelo pomembno je razložiti, zakaj uporaba branja v funkcijah namesto uporabe parametrov pogosto zmanjšuje uporabnost funkcije.

Prav tako dijakom razložimo, zakaj s pripravo funkcije, ki rezultat vrača, ne pa ga izpiše znotraj funkcije, posplošimo obnašanje kode.

MODULARNOST 210

CILJI

Dijak:

- : sestavi specifikacijo funkcije (določi namen funkcije in njene parametre),
- : sprogramira funkcijo glede na podano specifikacijo
- : razčleni program na manjše dele, in jih implementira v obliki funkcij,
- : razlikuje med izpisovanjem in vračanjem rezultatov funkcij.

STANDARDI ZNANJA

Dijak:

- » izdelava opis funkcije s tem, da opiše, kaj funkcija naredi in nad katerimi podatki deluje,
- » **vrednoti uporabo vhodnih parametrov funkcije** in razloži, zakaj branje znotraj funkcije namesto prenosa vrednosti preko parametrov pogosto zmanjšuje njeno uporabnost,
- » **vrednoti uporabo vračanja vrednosti funkcije (return)**. Razloži, zakaj s pripravo funkcije, ki rezultat vrača, ne pa ga izpisuje znotraj funkcije, praviloma posplošimo obnašanje kode (razlikuje med uporabo print in return),
- » na osnovi danega opisa funkcije zapiše ustrezno implementacijo in jo uporabi v rešitvi.

TERMINI

○ funkcija ○ klic funkcije ○ knjižnice funkcij ○ parametri ○ modul

DIDAKTIČNA PRIPOROČILA ZA SKUPINO CILJEV

Upošteujemo predvsem smiselni del splošnih priporočil za temo Algoritmi in programiranje.

Glede na pomembnost funkcij predvidevamo, da se bodo dijaki že v obveznem 70-urnem delu temeljito seznanili s funkcijami. V vsakem primeru pa je dobro, da ob ponovitvi tega znanja opozorimo na razliko med parametri (spremenljivke, ki služijo opisu delovanja funkcije), in argumenti (dejanske vrednosti parametrov, ki jih funkcija prejme ob klicu).

Kot je navedeno že v didaktičnih priporočilih za 70-urni del je dobro pokazati vsaj primere, kjer za argumente uporabimo sestavljene izraze, saj ti dijakom pomagajo razumeti, da med imeni parametrov in argumenti ni nujne povezave. Npr. Izračun diagonale pravokotnika z uporabo matematične funkcije sqrt:



```
c = math.sqrt(a*a + b*b)
```

načeloma enaka kodi

```
vso = a*a + b*b
c = math.sqrt(vso)
```

Zelo pomembno je ponovno razložiti in demonstrirati, zakaj je bolje, da podatkov v funkcijah praviloma ne beremo, ampak ustrezne vrednosti dobimo preko parametrov. Tak pristop prispeva k večji modularnosti, preglednosti in ponovni uporabnosti funkcij.

Prav tako dijakom razložimo, zakaj je funkcija, ki rezultat vrača, ne pa ga izpiše znotraj funkcije, uporabnejša od tiste, ki rezultat izpiše neposredno (s print)..

Razlog za to je, da funkcije, ki rezultat vračajo, omogočajo večjo prilagodljivost in ponovno uporabo v različnih kontekstih.

Ko funkcija rezultat vrača, lahko ta rezultat uporabimo za nadaljnje izračune, shranimo v spremenljivko ali pa ga po potrebi izpišemo znotraj drugih delov programa. Če pa funkcija rezultat neposredno izpisuje, je njen izhod omejen na takojšnje prikazovanje, kar zmanjšuje njeno uporabnost v situacijah, kjer bi bil potreben zgolj rezultat brez neposrednega izpisa.

S tem ločimo odgovornosti funkcije – vračanje podatkov (logika) in izpisovanje podatkov (prikazovanje). Takšna zasnova izboljšuje modularnost, olajša testiranje funkcij in povečuje berljivost ter vzdržljivost kode.

Predvsem naj bi bil pri obravnavi koncepta funkcij poudarek na vlogo funkcij pri delitvi na podprobleme: Demonstriramo (s "programiranjem v živo") prednosti postopka, ko na določenem mestu potrebujemo rezultat in za pridobitev slednjega le pokličemo (neobstoječo) funkcijo in nadaljujemo z razvojem. Temu potem sledi kasnejši razvoj te funkcije.

Modularnost, ki jo izražamo v obliki funkcij, je izredno močno orodje pri sestavljanju programov. Pomaga nam pri iskanju napak in preverjanju pravilnosti postopkov pri kompleksnejših algoritmihih (enostavneje je celovito preveriti dele kode - funkcije in naknadno preveriti še "zloženke" klicev funkcij kot pa preverjati celotno, monolitno kodo.



RAZVOJ PROGRAMOV 70

CILJI

Dijak:

○: avtentične probleme modelira in jih predstavi kot računalniško-informacijske probleme,

SC (4.5.3.1)

○: razvija programe s postopkom, ki vključuje načrtovanje, zapisovanje programa, izvajanje, preizkušanje in odpravljanje neskladnosti z modelom in načrtom,

SC (4.3.4.1)

○: upošteva lastna močna in šibka področja s področja programiranja, spremlja napredek in reflektira svoje delo.

SC (3.1.3.1)

STANDARDI ZNANJA

Dijak:

- » pri načrtovanju rešitve podanega problema opredeli zahteve in omejitve delovanja rešitve,
- » **program zapiše v skladu z načrtovanimi zahtevami,**
- » program preizkuša in *odpravlja neskladja z načrtom.*

TERMINI

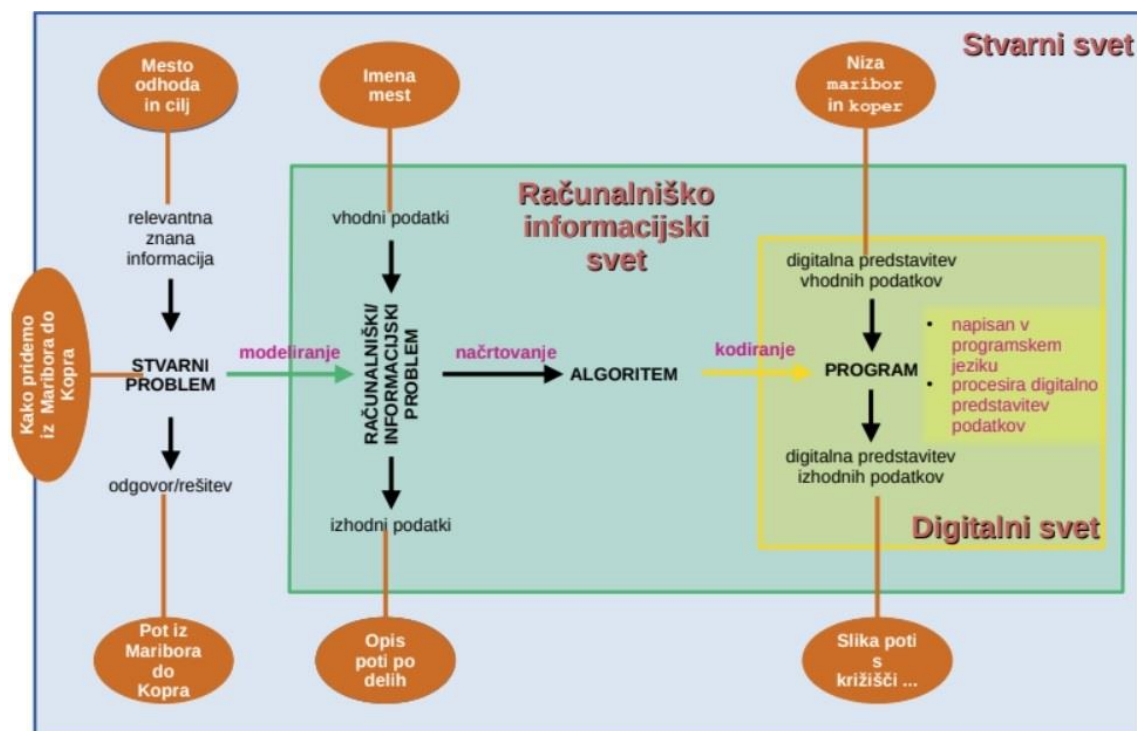
◦ problemski kriteriji ◦ funkcionalna specifikacija ◦ računalniško-informacijski problem

DIDAKTIČNA PRIPOROČILA ZA SKUPINO CILJEV

Upošteevamo predvsem smiselni del splošnih priporočil za temo Algoritmi in programiranje.

Ta skupina ciljev predstavlja glavni del obravnave teme, kjer se zaokroži celovito poznavanje algoritmov in programiranja.

Izhajamo iz v uvodnem delu predstavljenega prispevka Andreja Brodnika, Andrewa Csizmadie, Geralda Futscheka, Lidije Kralj, Violette Lonati, Petra Micheuza in Mattie Monge »Programming for All: Understanding the Nature of Programs (<https://arxiv.org/abs/2111.04887>)« in tam prikazane sheme »Od stvarnega problema do programa«.



Avtentični problem, ki ga dijaki rešujejo, je lahko povezan tudi s trajnostnim razvojem, npr.:

13. <https://aplikacijaun.zrss.si/api/link/12epzxh>
14. **Life on Land (Python)** – Tynker <https://aplikacijaun.zrss.si/api/link/az3qwps>
15. **Introduction to Python for Environmental Data Analysis** – Udemy <https://aplikacijaun.zrss.si/api/link/ton01f3>
16. **Environmental Science Python Projects** – Research1.fun <https://aplikacijaun.zrss.si/api/link/65pjf6o>

in s preostalimi področji skupnih ciljev.

Pri razvoju programov uporabljamo različne tehnike organizacije pouka, kot so med drugim uporaba dela v skupini, programiranje v parih, igranje vlog (*naročnik programa*, *programer*, *preizkuševalec*), agilni pristop kot metoda razvoja programske opreme in vloge v tem postopku.

Dodatni primeri:

- » iskanje trojic števil na določenem intervalu z lastnostjo $a^2 + b^2 = c^2$ (s trojno oz. z dvojno zanko); iskanje v urejenem seznamu (linearno oz. dvojiško); iskanje parov elementov urejene tabele, ki imajo določeno vsoto; urejanje tabele ...
- » Preverjanje praštevil: Ali je število 97 praštevilo? Prva metoda preverja vse delitelje od 2 do 96, druga metoda pa le do $\sqrt{97} \approx 9.8$, torej do 9.

- » Izpis praštevil na intervalu: Najdi vsa praštevila med 1 in 100. Osnovni pristop: za vsako število preveri vse delitelje. Izboljšava 1: preskoči soda števila (razen 2). Izboljšava 2: preverjaj delitelje le do korena. Eratostenovo sito: označi večkratnike 2, 3, 5, 7... in obdrži neoznačene.
- » Iskanje trojic števil z lastnostjo $k^2 + m^2 = n^2$: Najdi vse trojice (k, m, n) med 1 in 50, kjer velja $k^2 + m^2 = n^2$ (npr. $3^2 + 4^2 = 5^2$). Trojna zanka: preveri vse kombinacije k, m, n . Dvojna zanka: za k in m izračunaj $n = \sqrt{k^2 + m^2}$ in preveri, ali je celo število.
- » Iskanje v urejeni tabeli: V seznamu [1, 3, 7, 12, 18, 23, 29, 35, 42] poišči število 23. Zaporedno iskanje: preveri 1, 3, 7, 12, 18, 23 (6 korakov). Bisekcija: preveri $18 \rightarrow 29 \rightarrow 23$ (3 koraki).
- » Iskanje parov elementov z določeno vsoto: V seznamu [2, 4, 6, 8, 10, 12] najdi dva števila, ki se seštejeta v 14. Osnovni pristop: preveri vse pare. Pametni pristop: za vsako število a poišči $b = 14 - a$.
- » Urejanje seznama: Uredi seznam [64, 34, 25, 12, 22, 11, 90]. Urejanje z mehurčki: primerja sosednja elementa. Hitro urejanje: razdeli glede na osrednji element.



RAZVOJ PROGRAMOV 210

CILJI

Dijak:

- : vrednoti avtentične izzive kot priložnosti za ustvarjanje vrednosti zase in za druge,
- : modelira avtentični problem, za njegovo rešitev razvija algoritem in ga kodira v delujoč program,
- SC (5.1.1.1 | 5.1.2.2)
- : izpostavlja čas izvajanja kot eno ključnih meril za primerjavo različnih programov, namenjenih reševanju istega problema.

STANDARDI ZNANJA

Dijak:

- reši avtentični problem, ki naslavlja želje in potrebe okolja, tako da:
 - » postavi ustrezen model rešitve,
 - » definira problemske kriterije in omejitve,
 - » razvije ustrezen algoritem in program,
 - » preveri, ali so kriteriji in omejitve izpolnjeni;
- prikaže primer problema in vsaj dveh programskih rešitev tega problema, ki se razlikujeta glede na čas izvajanja.

TERMINI

- problemski kriterij ◦ model rešitve ◦ funkcionalna specifikacija ◦ računalniško informacijski problem

DIDAKTIČNA PRIPOROČILA ZA SKUPINO CILJEV

Upoštevamo predvsem smiselni del splošnih priporočil za temo Algoritmi in programiranje.

Pri razvoju programov obravnavamo tudi različne tehnike iskanja napak in razhroščevanja. Samo razhroščevanje vgrajujemo v razlago in aktivnosti že pri obravnavi drugih skupin ciljev. Pomembna tehnika pri iskanju tako napak kot tudi pomena same zapisane kode je sledenje izvajanja vrstic kode. Pri tem lahko uporabljamo v samo razvojno okolje vgrajene možnosti (npr. Thonny: debug current script) oziroma specializirana okolja (npr. <https://pythontutor.com>).



Avtentični problem, ki ga dijaki rešujejo, je lahko povezan tudi s trajnostnim razvojem (<https://aplikacijaun.zrss.si/api/link/12epzxh>) in preostalimi področji skupnih ciljev. Pri razvoju rešitve se lahko dijaki delijo v skupine in s tem krepijo sodelovalno delo (vsak prispeva svoj del mogoče skupne rešitve) in se učijo en od drugega ter si delijo vloge (naročnik, razvijalec, programer, preizkuševalec ...).

Učitelj naj načrtuje tudi razvijanje podjetnosti dijakov v smislu načrtovanja kratkoročnih, srednjeročnih in dolgoročnih ciljev, opredelitve prednostnih nalog in priprave načrta.  (5.3.2.1)

V sklopu tega dela obravnavamo tudi probleme, za katere lahko poiščemo več algoritmov, ki ta problem rešijo, a to storijo različno hitro (npr. preverjanje, ali je n praštevilo, s preverjanjem deliteljev do $n - 1$ v primerjavi s preverjanjem do $\sqrt{n}$). Prav tako je problem izpisa praštevil na določenem intervalu dodaten primer problema, ki ga je mogoče rešiti na zelo različne načine (npr. najprej se sprehodimo po celotnem intervalu in za vsako kandidatno število preštejemo njegove delitelje; nato ugotovimo, da lahko kandidatno število proglasimo za sestavljeno število takoj, ko odkrijemo, da je z nekim številom deljivo; zatem ugotovimo, da se lahko /če odmislimo praštevilo 2/ tako pri kandidatnih številih kot pri deliteljih omejimo na liha števila; potem ugotovimo, da se lahko v zanki po deliteljih sprehodimo samo do korena kandidatnega števila; na koncu pa spoznamo še Eratostenovo sito – algoritem, ki je časovno še učinkovitejši, četudi na račun večje porabe pomnilnika).

Dodatni primeri, kjer lahko poiščemo različno hitre algoritme za rešitev: iskanje trojic števil na določenem intervalu z lastnostjo $k^2 + m^2 = n^2$ (primerjamo uporabo trojne in dvojne zanke); iskanje v urejenem seznamu (zaporedno ali z bisekcijo); iskanje parov elementov urejene tabele, ki imajo določeno vsoto; urejanje tabele ...

IZBIRNI CILJI 210

CILJI

Dijak:

- l: analizira lastnosti problema, ki ga lahko potem reši z določenim standardnim pristopom (rekurzija, uporaba strategij, kot sta deli in vladaj in sestopanje),*
- l: interpretira osnovne pojme objektnega programiranja, kot so objekt, razred, lastnost in enkapsulacija, ter jih uporablja pri reševanju problemov,*
- l: raziskuje podatkovne strukture, kot so slovarji, graf, drevo in množica, ter spoznava tipične operacije nad njimi,*
- l: izboljša program glede možnosti implementacije, berljivosti ali zmogljivosti,*
- l: uporablja orodje in okolje za razvoj programov.*

STANDARDI ZNANJA

Dijak:

- demonstrira uporabo rekurzije kot pogostega pristopa pri razvoju algoritma;
- demonstrira nekaj tipičnih strategij pri razvoju algoritmov, kot sta deli in vladaj ter sestopanje;
- na primerih uporabi ter interpretira osnovne pojme objektnega programiranja, kot so objekt, razred, lastnost in enkapsulacija;
- prikaže primer problema, kjer uporabi osnovne prijeme objektno orientiranega programiranja (OOP);
- prikaže primer standardnih razredov (ko sta na primer random in turtle), tako da:
 - » ustvari nekaj primerov objektov ustreznih razredov,
 - » opiše lastnosti in metode, ki so na voljo v teh razredih;
- prikaže primer razvoja modula kot kombinacije podatkov in podprogramov, tako da načrtuje postopek reševanja problema:
 - » ustvari ustrezen primer razreda (ali več njih),
 - » z uporabo lastnosti in metod objektov reši problem;
- primerja tipične operacije in uporabo najpomembnejših podatkovnih struktur, kot so slovarji, graf, drevo in množica, ter navede tipične operacije nad njimi;
- analizira učinkovitost, berljivost kode in možnost nadgradnje vsaj dveh programov, ki rešita isti problem;
- demonstrira uporabo orodja za razvoj programov, tako da:

- » vključuje obstoječo kodo in jo po potrebi prilagodi,
- » izvede testiranje posameznih delov programa in programa kot celote,
- » odpravi napake in neskladja z načrtom.

TERMINI

- problemski kriterij
- funkcionalna specifikacija
- računalniško informacijski problem

DIDAKTIČNA PRIPOROČILA ZA SKUPINO CILJEV

Upošteujemo predvsem smiselni del splošnih priporočil za temo Algoritmi in programiranje.

Pri rekurziji se (zagotovo vsaj na začetku, običajno pa tudi nasploh) izogibamo demonstraciji izvajanja s skladom rekurzivnih klicev. Na rekurzivni klic moramo gledati kot na klic »običajne« funkcije – pomembno je, kakšen rezultat klic funkcije vrne in ne kako do rezultata pride. To, da kličemo isto funkcijo, je le »naključje«.

Pri vpeljavi rekurzije se ob vpeljavi pojma izogibamo uporabi primerov, kjer je »naravni« pristop pravzaprav zanka (fakulteta, Fibonaccijeva števila, vsota naravnih števil do n ...). Raje uporabimo zglede, kjer je rekurzija »naravna« (problem hanojskih stolpičev, iskanje k -tega največjega števila ...), in take, kjer je iterativna rešitev bolj »skrta« (npr. izračun y^n).

Pri vpeljavi je smiselno uporabiti pristop računalništva brez računalnika. Algoritme, kot so urejanje z zlivanjem, hitro razvrščanje (quick sort) in iskanje k -tega po velikosti, lahko lepo prikažemo na problemu razporejanja predmetov po teži s pomočjo nagibne tehtnice. Prav tako so dostopni tudi modeli za problem hanojskih stolpičev, kjer rekurzivni klic izvedemo kot »goljufanje«, ko hkrati prenesemo več obročev in tolmačimo, da je to isti problem, le z drugače označenimi stolpi in drugačnim številom obročev.

Uporabni zgledi za rekurzijo so fraktalni objekti, kot so Kochova črta, trikotnik Sierpinskega, Mengerjeva spužva, itd ...

Pri obravnavi OOP je pomembno predvsem upoštevati enkapsulacijo kot temeljni pristop pri tem načinu programiranja. Demonstriramo primer, ko podatke objektov določenega razreda predstavimo na en način in potem spremenimo to predstavitev v drug način ter pokažemo, da se s stališča uporabe objektov tega razreda ni nič spremenilo.

Posebej velja biti pozoren pri vpeljavi OOP, kadar se kot programski jezik uporablja python. Tu je namreč uporaba enkapsulacije prepuščena programerju (razlika med lastnostjo in »navadno« spremenljivko, kjer hranimo podatke o objektu). Žal je zelo veliko gradiv za OOP v pythonu napisanih tako, da se uporablja neposredni dostop do podatkov objekta in enkapsulacija dejansko ni udejanjena.

Ustrezen uvod v OOP je v primeru uporabe pythona predstavitev želje grafike, kjer želvo predstavimo kot objekt z lastnostmi (položaj, smer, stanje peresa). Nad tem objektom izvajamo metode, ki spremenijo stanje objekta (lastnosti), ob tem pa lahko imajo še dodaten učinek (puščanje sledi peresa).

RAČUNALNIŠKA OMREŽJA

OBVEZNO

OPIS TEME

Dijaki se seznaniijo z vlogo računalniških omrežij pri povezovanju ljudi in naprav ter s tem povezanih varnostnih in zasebnostnih vprašanj. Spoznavaajo, kako sporočila potujejo preko omrežij, in se seznaniijo z osnovnimi principi delovanja interneta, vključno z osnovami protokolov, kot je TCP/IP. Znajo opisati različne načine varovanja zasebnosti, podatkov in naprav.

DIDAKTIČNA PRIPOROČILA ZA TEMO

Kot uvod v posamezno skupino ciljev lahko učitelj predvaja kratek videoposnetek. Pred predvajanjem učitelj opozori dijake, na kaj naj bodo pozorni, vsebino lahko ob zaustavitvi posnetka tudi dodatno obrazloži. V prvem letniku (70 ur) je za posamezni cilj kot uvod v tematiko mogoče pogledati kratek video s seznama: <https://aplikacijaun.zrss.si/api/link/rd6imjq>. Pri predvajanju posnetkov lahko vklopimo samodejne podnapise v slovenščini. Za primer medpredmetnega povezovanja pa si lahko ogledamo iste posnetke z angleškimi podnapisi.

Po ogledu posnetka sledi kratka dejavnost dijakov, kjer se glede na videno sami preizkusijo v vlogi ene izmed naprav v omrežju v postopku pošiljanja paketa od ene končne naprave do druge in skušajo odgovoriti, zakaj neka naprava dela po nekem postopku oz. kateri problem rešuje.

Učitelj pri pouku uporabi tudi orodje za simulacijo omrežij in demonstrira prenos paketov v omrežju. Demonstrira tudi povezovanje naprav v lokalno omrežje, pri čemer uporabi ročne nastavitve IP ali samodejno pridobivanje nastavitvev IP.

Za učiteljeve demonstracije se za simulacijo delovanja računalniških omrežij priporoča uporaba ustreznih orodij (na primer Packet Tracer, GNS3, Boson NetSim, Cloonix, NetSim, CORE, Netkit ...).

Dijakom za potrebe samostojnega dela pripravimo navodila za namestitvev in uporabo simulacijskega orodja. Učitelj za utrjevanje doma pripravi osnovne naloge.

Gradivo:

Spletni učbenik: <https://aplikacijaun.zrss.si/api/link/xci2gf>

Pri skupinah ciljev, ki imajo pripisano »210«, lahko dijaki v parih ali tudi posamezno na računalnikih rešujejo krajše naloge v simulacijskem orodju. Naloge so lahko namenjene tako usvajanju novih znanj kot tudi utrjevanju snovi. Dijaki lahko uporabljajo simulacijsko orodje; če je možnost, pa tudi poskusijo računalnike povezati v LAN (lokalno omrežje) in WLAN (brežžično lokalno omrežje), pri čemer uporabijo ročne nastavitve IP ali samodejno pridobivanje nastavitvev IP.

Uporabi se lahko tudi protokolni analizator za analizo omrežnega prometa ter omrežna orodja operacijskega sistema, kot so ukazi v ukazni lupini (npr. ping, ipconfig, tracert) in grafični vmesniki za omrežne nastavitve.

NAMEN IN ZGRADBA OMREŽIJ 70

CILJI

Dijak:

○: raziskuje namen in zgradbo računalniških omrežij.

STANDARDI ZNANJA

Dijak:

- » pojasni namen računalniškega omrežja,
- » našteje sestavne elemente računalniškega omrežja (omrežne naprave, končne naprave, povezave),
- » analizira namen računalniškega omrežja za različne storitve,
- » primerja razliko med lokalnim in globalnim omrežjem.

TERMINI

- računalniško omrežje ◦ internet ◦ splet ◦ omrežna naprava ◦ končna naprava ◦ omrežne povezave
- lokalno omrežje ◦ globalno omrežje ◦ omrežne storitve

DIDAKTIČNA PRIPOROČILA ZA SKUPINO CILJEV

Učitelj razloži pojem računalniškega omrežja in našteje sestavne elemente računalniškega omrežja (omrežne naprave, končne naprave, povezave).

Predvaja posnetek primera potovanja paketa preko medmrežja, tj. posnetek, ki prikazuje, kaj se zgodi od trenutka, ko vpišemo spletni naslov v brskalnik, do trenutka, ko se v brskalniku prikaže vsebina.

Dijaki razmislijo in odgovorijo, kateri pogoji so potrebni, da lahko govorimo o računalniškem omrežju. Kdaj je računalnik povezan v omrežje? Ali lahko poleg računalnikov povežemo v omrežje tudi druge naprave, ki jih ljudje uporabljamo.

Zakaj vsaka naprava, ki jo povežemo v omrežje, potrebuje naslov? Ali imajo naprave v istem omrežju podoben naslov?



Z uporabo pripomočkov demonstrirajo razumevanje, kako dve lokalni omrežji povezati v medmrežje in poslati sporočilo med dvema končnima napravama. Uporabijo lahko npr. legokocke različnih barv za vsak element omrežja in vrvico za povezavo ali npr. barvne kape iz papirja: ko si jo dijak nadene, predstavlja en element omrežja. Po navodilu učitelja dijaki vzpostavijo želeno obliko omrežja.

Za lažje razumevanje vlog posameznih sestavnih elementov se lahko primerja računalniško omrežje s storitvijo klasične pošte, ki jo dijaki razumejo in poznajo. Razmislijo in odgovorijo, kakšna je vloga pismonoše (vloga mrežnih naprav v lokalnem omrežju) in zaposlenih na poštah (vloga mrežnih naprav v medmrežju) ter kakšne so možnosti pošiljanja pošte z uporabo različnih tehnologij oz. transportnih sredstev. Dijaki aktivno sodelujejo v aktivnosti (npr. barvne kape iz papirja) kot elementi omrežij in v učilnici izvedejo aktivnost prenosa sporočila med končnima napravama iz različnih omrežij (kot prenos pisma med dvema osebama iz različnih krajev).

Učitelj lahko pripravi demonstracijo delovanja omrežja v simulacijskem okolju. Preko primera omrežja v simulacijskem okolju se prikaže vloga končnih in mrežnih naprav ter povezav. Naloga v simulacijskem okolju je lahko sestavljena tako, da dijake vodi skozi postavitev in povezovanje nekaj končnih naprav na mrežno napravo, čemur sledi preizkus pošiljanja paketov.

Učitelj razloži namen računalniškega omrežja skozi možnosti uporabe storitev.

Dijaki razmislijo, katere storitve uporabljajo oz. preko katerih orodij uporabljajo omrežja (brskalnik, odjemalec e-pošte, orodje za videokonference, shranjevanje in deljenje datotek preko oblčnih storitev ...).

Učitelj razloži razliko med lokalnim in globalnim omrežjem.

Dijaki razmislijo o potrebi po hierarhični ureditvi omrežij v primerjavi s storitvijo klasične pošte (lokalni pismonoša, prenos pošte med mesti/državami). Zakaj imamo v storitvi klasične pošte vloge ločene (lokalni pismonoša / poštni uslužbenci v stavbah lokalnih pošt / poštni uslužbenci v stavbah pošt večjih mest v državi)?

Lahko razloži tudi vzporednico z elektroenergetskim omrežjem – kje je meja med zunanjim omrežjem (distributer) in omrežjem doma / v šoli / v podjetju. Katera naprava je meja med obema omrežjema?



NAMEN IN ZGRADBA OMREŽIJ 210

CILJI

Dijak:

- raziskuje osnovne koncepte omrežnih storitev, uporabo protokolov med entitetami v omrežju in strukturo ter funkcijo protokolnega sklada,
- primerja različne topologije omrežij,
- analizira vlogo omrežnih naprav in utemeljuje izbor ustreznih medijev za prenos podatkov,
- pojasnjuje dodeljevanje naslovov napravam v omrežju in njihovo strukturo.

STANDARDI ZNANJA

Dijak:

- » opiše standardne aplikacijske storitve, ki jih uporabljamo v omrežjih,
- » utemelji pomembnost standardizacije storitve,
- » pojasni razliko med storitvami v omrežju "enakovrednih" (peer-to-peer) in "odjemalec-strežnik" (client-server),
- » opiše koncept protokola med členoma entitetnega para določene storitve,
- » razloži namen protokolnega sklada in opiše vlogo posameznih slojev,
- » analizira pomen, vlogo in funkcionalnost omrežnih naprav (stikala, usmerjevalniki, dostopovne točke),
- » opiše zgradbo in prednosti različnih topologij omrežij,
- » pojasni dodeljevanje enoličnega naslova, ki sestoji iz naslova omrežja in naslova naprave v tem omrežju,
- » primerja omrežne medije (brežžične povezave, UTP, optika ...) in za podan primer izbere najustreznejšega ter utemelji izbiro.

TERMINI

○ protokolni sklad ○ omrežni protokoli ○ topologija ○ stikalo ○ usmerjevalnik ○ dostopovna točka ○ enolični omrežni naslov ○ UTP ○ optika ○ omrežni mediji

DIDAKTIČNA PRIPOROČILA ZA SKUPINO CILJEV

Učitelj razlaga osnovne koncepte omrežnih storitev, uporabo protokolov med entitetami v omrežju in strukturo ter funkcijo protokolnega sklada:

Dijak utemelji potrebo po standardizaciji storitev kot potrebo za globalno komunikacijo. Razloži potrebnost uvedbe protokolov v primerjavi s storitvami klasične pošte in drugimi storitvami – zakaj so pravila in postopki, ko vstopiš v stavbo pošte/banke/..., zakaj morajo pravila oz. postopke upoštevati pismonoše in drugi poštni uslužbenci.

Učitelj lahko izvede demonstracijo s pomočjo simulacijskega orodja, kjer prikaže uporabo ene izmed storitev (npr. prenos spletne strani) in pri tem naredi uvod v koncept protokolnega sklada. Dijak preko pripravljene vaje v simulacijskem orodju pregleduje, kaj se dogaja na posameznih plasteh, in pri tem poskuša razložiti namen protokolnega sklada, kjer loči:

storitve aplikacijske plasti (uporaba različnih aplikacij),

storitve prenosne plasti – navede pravilno storitev na prenosni plasti za potrebe neke aplikacijske storitve,

storitev omrežne plasti – razloži pomen enotnosti omrežne plasti v globalnem omrežju in neodvisnost te plasti od tehnologije prenosa,

storitve oz. vlogo dostopovne plasti, kjer razloži pomen standardizacije na ravni uporabljenih tehnologij prenosa.

Učitelj pojasnjuje vlogo mrežnih naprav in utemeljuje izbor ustreznih medijev za prenos podatkov.

Preko simulacijskega orodja se lahko prikaže vloga stikala, usmerjevalnika in dostopne točke. Poleg demonstracije učitelj pripravi nalogo za dijake, kjer bodo ob spremljanju dogajanja v simulacijskem orodju sami spoznali namen in delovanje različnih mrežnih naprav.

Dijak pojasni, zakaj želimo imeti možnosti pošiljanja paketov po različnih poteh za zanesljivost delovanja interneta in pri tem vlogo usmerjevalnika oz. usmerjevalnih protokolov. Učitelj demonstrira delovanje omrežja, kjer ob izpadu ene povezave paketi potujejo do cilja po drugi poti.

Dijaki raziščejo in ugotovijo tip omrežnega medija v računalniški učilnici, primerjajo s teoretičnimi značilnostmi medija in razmislijo o utemeljitvi izbire (primerjava med UTP in optiko). Mogoče je v katerem delu šole tudi optična napeljava in si jo lahko ogledajo. Preverijo ponudbo tehnologij dostopa do internetnega ponudnika in preverijo, kakšno vrsto dostopa imajo doma. Razmislijo o prednostih in slabostih uporabe brezžičnih povezav. Kakšen domet ima njihovo domače brezžično omrežje?

Dijaki utemeljujejo izbor naprave in medija za neko praktično situacijo (doma – razširitev omrežja v drugo nadstropje ...).

Dijaki lahko uporabijo orodja/ukaze (ping, tracert oz. traceroute) za testiranje povezave do določene naprave v omrežju.

Dijaki na spletu poiščejo shemo Arnesovega hrbtničnega omrežja, z uporabo ukaza tracert/traceroute praktično raziskujejo topologijo omrežja ponudnika internetnih storitev.

PRENOS SPOROČIL 70

CILJI

Dijak:

○: pojasnjuje razlog deljenja sporočila na manjše dele.

STANDARDI ZNANJA

Dijak:

- Opiše pot sporočila od ene naprave do druge in pri tem:

- » razloži razlog deljenja sporočila na manjše dele,
- » razloži, kaj je treba dodeliti posameznim delom sporočila, da jih bo mogoče dostaviti do ciljne naprave,
- » razloži, kako je dosežena odpornost proti izpadom dela omrežja.

DIDAKTIČNA PRIPOROČILA ZA SKUPINO CILJEV

Učitelj pojasnjuje prenos sporočil v računalniškem omrežju.

Učitelj lahko predvaja posnetek primera potovanja paketa preko medmrežja, kjer je poudarek na uporabi naslovov.

Po ogledu posnetka se izvede aktivnost dijakov, kjer z barvami papirnatih kap dijaki prevzamejo vlogo končnih in mrežnih naprav, pri čemer je poudarek na pomenu naslovov naprav v omrežjih. Izvedejo prenos sporočila od ene končne naprave do druge, pri čemer si sporočilo izmenjajo tudi dijaki z vlogo mrežnih stikal in usmerjevalnikov. Učitelj jim je v pomoč pri vrstnem redu postavitve. Dijaki razmislijo in odgovorijo, kako se naslovi uporabljajo v storitvi klasične pošte – kateri del naslova in zakaj samo ta del je pomemben za lokalnega pismonošo? Zakaj za lokalnega pismonošo ni dovolj zapis naslova na pisemski ovojnici/paketu? Zakaj je za poštne uslužbence pomemben samo en del naslova na ovojnici in kateri?

Dijaki razmislijo o neodvisnosti posameznih naprav pri odločanju o prepošiljanju paketov v omrežju, tako da primerjajo odločanje poštne uslužbenec pri prepošiljanju pošte. Ali je nujno, da grede pošiljke vedno po isti poti? Učitelj lahko poda primer pošiljanja pošiljke v tujino. Dijaki razmislijo in odgovorijo, ali je nujno, da posamezni poštni uslužbenec pozna celotno pot pošiljke. Pri tem sklepajo o tem, kako so posamezna vozlišča v omrežju odvisna od sosednjih, in o potrebnosti poznavanja celotne poti paketov.

Dijaki lahko razmislijo in poskušajo razložiti, zakaj je treba deliti sporočila v pakete, kako je to v realnem svetu – zakaj imamo omejitve pri prenosu stvari preko pošte / paketne dostave / ljudi v avtomobilih ali avtobusih.

Razmislijo, zakaj nimamo avtobusa za 500 potnikov?

PRENOS SPOROČIL 210

CILJI

Dijak:

- : analizira redundantnost omrežij,
- : raziskuje kako se zagotavlja zanesljivost dostavljanja sporočil,
- : analizira možnosti naslavljanja naprav.

STANDARDI ZNANJA

Dijak:

- pojasni, kako redundantnost v omrežjih in odločanje o usmerjanju paketov izboljšujeta zanesljivost delovanja interneta;
- **analizira vlogo usmerjanja v omrežju** pri zagotavljanju zanesljivosti delovanja interneta;
- **razloži, kako se omrežna naprava odloči o usmerjanju paketa;**
- poišče omrežne nastavitve naprave in **analizira vlogo posameznih elementov (IP naslov, omrežna maska, privzeti prehod, strežnik domenskih imen (DNS))**;
- razloži pomen storitve dinamičnega dodeljevanja omrežnih nastavitev;
- opiše razloge, zakaj se naprava ne more priključiti v omrežje, tako da  (4.5.1.1):
 - » preveri fizično povezavo,
 - » preveri naslavljanje naprave,
 - » preveri povezavo do prejemnika (ping, tracert).

TERMINI

- mrežna maska ○ privzeti prehod ○ DNS ○ usmerjanje ○ naslov IP

DIDAKTIČNA PRIPOROČILA ZA SKUPINO CILJEV

Dijaki razmislijo in pojasnijo podobnost cestnega omrežja z računalniškim omrežjem (križišča, hitrosti, vzporedni pasovi, alternativne poti, zgostitve in zastoji ...). Na podlagi podobnosti s cestnim omrežjem sklepajo o vplivu hierarhične zgradbe (lokalne ceste, regionalne ceste, avtoceste), možnosti uporabe alternativnih poti in sprotnega odločanja o izbiri poti glede na informacije o stanju na poti (nesreče, zastoji) na zanesljivost

delovanja omrežja. Pri tem prepoznajo koncept usmerjevalnika (križišče) in vlogo usmerjanja (navodila navigacijske naprave, uporaba npr. Googlovih Zemljevidov) pri zagotavljanju zanesljivosti delovanja interneta.

Dijaki razmislijo in poskušajo razložiti, zakaj potrebujemo naslove (MAC, IP, vrata TCP/UDP)? Pri tem jih vodi učitelj, ki jim pomaga ustvariti analogijo med naslovi, ki se uporabljajo pri prenosu paketov v računalniških omrežjih, in naslovi pri storitvi klasične pošte.

Učitelj vodi dijake, da v nastavitvah operacijskega sistema poiščejo omrežne nastavitve. Zapišejo naslove: MAC, IP. Učitelj razloži, zakaj imamo poleg IPv4 tudi naslove IPv6, in razloži vlogo posameznih elementov mrežnih nastavitvev (naslov IP, maska, privzeti prehod, strežnik DNS).

Z uporabo simulacijskega orodja dijaki preizkušajo pošiljanje paketov oz. uporabo storitev ob izbrisu nastavitvev (npr. manjka naslov strežnika DNS, manjka omrežni prehod, manjka naslov IP).

Preverijo možnost uporabe dinamičnih nastavitvev v učilnici, doma. Razmislijo, zakaj bi v določenem primeru želeli imeti to možnost oz. zakaj je ta način uporaben. Razmislijo in odgovorijo, zakaj pa v drugih primerih želimo imeti statične nastavitve IP. Razmislijo, zakaj je bil vpeljan zasebni naslovni prostor – posebni naslovi IP, ki jih uporabljamo v domačih omrežjih, lahko tudi kot interni naslovni prostor podjetij.

Učitelj pripravi primer omrežja v učilnici ali v simulacijskem orodju z napakami v postavitvi in nastavitvah (napačno povezan vodnik, napačno nastavljeni naslovi ...). Dijaki preverijo fizične povezave, nastavitve IP na končnih napravah in poskušajo najti napake.

Uporabljajo orodja/ukaze (ping, tracert) za testiranje povezave do določene naprave v omrežju.

Učitelj lahko pripravi demonstracijo nastavitve domačega brezžičnega usmerjevalnika. Dijaki opazujejo učitelja, ki nastavlja osnovne nastavitve domačega brezžičnega usmerjevalnika.

VARNOST 70

CILJI

Dijak:

○: analizira načine varovanja zasebnosti in podatkov pred nepooblaščenim dostopom,

SC (4.4.1.1 | 4.4.2.1)

○: vrednoti varno in odgovorno ravnanje na spletu ter pomembnost skrbi za lastno zdravje in zdravje drugih.

SC (3.2.4.2)

STANDARDI ZNANJA

Dijak:

pojasni pojme: identifikacija, avtentikacija in avtorizacija.

TERMINI

○ identifikacija ○ avtorizacija ○ avtentikacija (overovljenje)

DIDAKTIČNA PRIPOROČILA ZA SKUPINO CILJEV

Dijak pojasni pojme identifikacija, avtentikacija in avtorizacija skozi postopek prijave z uporabniškim računom (npr. z identiteto AAI), tako da:

- » razume, da obstajajo različni načini avtentikacije za različne storitve in da niso vsi enako dobri; na internetu poišče prijavne zaslone za različne storitve in komentira zahtevane podatke;
- » prepozna različne tipe osebnih podatkov (tekstovni, fotografije, ime in priimek, kraj bivanja, številka dijaške izkaznice, EMŠO, slika osebnega dokumenta ...);
- » pri tem osebne podatke opredeli kot bolj ali manj občutljive;
- » razume pomen izbire dobrega, »močnega« gesla in kako si ga lahko izmisli; predlaga več dobrih gesel;
- » razume, da se varnost prijave precej izboljša z uporabo večfaktorske avtentikacije; našteje nekaj načinov, s katerimi ponudniki to omogočajo (SMS, e-pošta, aplikacija OTP z žetoni itd.);
- » opiše tveganja pri objavi osebnih podatkov (vključno s hrambo podatkov uporabniškega profila pri ponudniku storitve) in pozna mogoče načine zlorabe;
- » prepozna bolj in manj varne storitve glede na raven varnosti, ki jo ponujajo, glede na podatke, ki jih zahtevajo pri prijavi, ter glede na državo in zakonodajo ponudnika (npr. EU ali Kitajska, Rusija, ZDA).

Dijakom je na voljo naslednje poglavje v e-učbeniku za informatiko:

<https://aplikacijaun.zrss.si/api/link/kones9k>.

Dijaki naj pri temi Varnost razvijajo občutljivost za moralna vprašanja ter sposobnosti, da o njih razmišljajo skupaj z drugimi in s tem razvijajo skupni cilj s področja Jezik, državljanstvo, kultura in umetnost?  (1.2.2.2)

Pri načrtovanju dejavnosti za razvijanje kompetenc 4.1 Skrb za varnost naprav

(<https://aplikacijaun.zrss.si/api/link/hy2l3ly>) in 4.2 Varovanje osebnih podatkov in zasebnosti

(<https://aplikacijaun.zrss.si/api/link/2byqenc>) učitelju lahko pomaga okvir DigComp 2.2 z opisi znanj, spretnosti in stališč za posamezno kompetenco.

VARNOST 210

CILJI

Dijak:

○: izvaja ukrepe varovanja zasebnosti, podatkov in naprav.

● SC (4.4.1.1 | 4.4.2.1 | 1.2.1.5)

STANDARDI ZNANJA

Dijak:

- pojasni zakrivanje pomena podatkov, zagotavljanje celovitosti podatkov in zagotavljanje istovetnosti ter pri tem:

» razloži razliko med simetričnim in asimetričnim šifriranjem,

» razloži odnos med javnim in zasebnim ključem,

» uporabi certifikat za podpis datoteke,

» uporabi certifikat za zakrivanje pomena vsebine (šifriranje),

» uporabi orodje za preverjanje celovitosti podatkov;

- opiše razloge, načine, strukturo in posledice kibernetских napadov;

- razloži ukrepe za varovanje naprav preko fizičnega dostopa ali/in dostopa preko omrežja.

TERMINI

○ certifikat ○ ključ ○ kriptografija ○ simetrično šifriranje ○ javni ključ ○ zasebni ključ ○ celovitost ○ zakrivanje

DIDAKTIČNA PRIPOROČILA ZA SKUPINO CILJEV

Dijaki kot pripravo na uro ali na začetku učne ure dobijo nalogo pregledati spletno stran <https://www.cert.si>, kjer najdejo poročilo o kibernetiski varnosti in ga pregledajo. Na podlagi prebranega v parih/skupinah razmislijo in opišejo razloge kibernetiskega napada in oblike oškodovanja žrtve, opišejo načine, strukturo in posledice kibernetских napadov.

Zakaj moramo naprave varovati pred poskusi dostopa, tudi če so samo poizvedovanja? Kako se v realnem svetu odzovemo, ko opazimo, da se pred podjetjem/šolo/doma gibljejo sumljive osebe? Dijak poskuša na spletu najti informacije o možnosti uporabe splošnih orodij in opiše primere zbiranja podatkov o šibkih točkah sistema in načine varovanja pred tem.

Razloži ukrepe za varovanje naprav preko fizičnega dostopa ali/in dostopa preko omrežja – zakaj moramo naprave (računalnike, mrežne naprave) tudi fizično zavarovati pred nepooblaščenim dostopom? Zakaj uporabljamo požarni zid, antivirusne programe, kaj je namen posameznih orodij? V operacijskem sistemu najde nastavitve požarnega zidu in protivirusnega programa.

Razloži, kako izvedemo identifikacijo, avtentikacijo in avtorizacijo.

Izvedba dejavnosti: <http://vidra.si/javna-gesla/Javna%20gesla.pdf> – dijak šifrira podatke in spozna koncept javnega in zasebnega ključa. Učitelj po izvedbi aktivnosti predstavi koncept digitalnega potrdila (certifikata) / usmeri dijake na spletno gradivo.

Pri delu s certifikati lahko dijaki uporabijo odprtokodne certifikate, kot je na primer certifikat PGP.

Dijak uporabi programsko orodje za šifriranje podatkov in orodje za preverjanje celovitosti podatkov. Uporabi lahko tudi orodje za zajem mrežnih paketov ter primerja šifrirano in nešifrirano sporočilo v smislu nevarnosti (kraja podatkov, gesla ...).

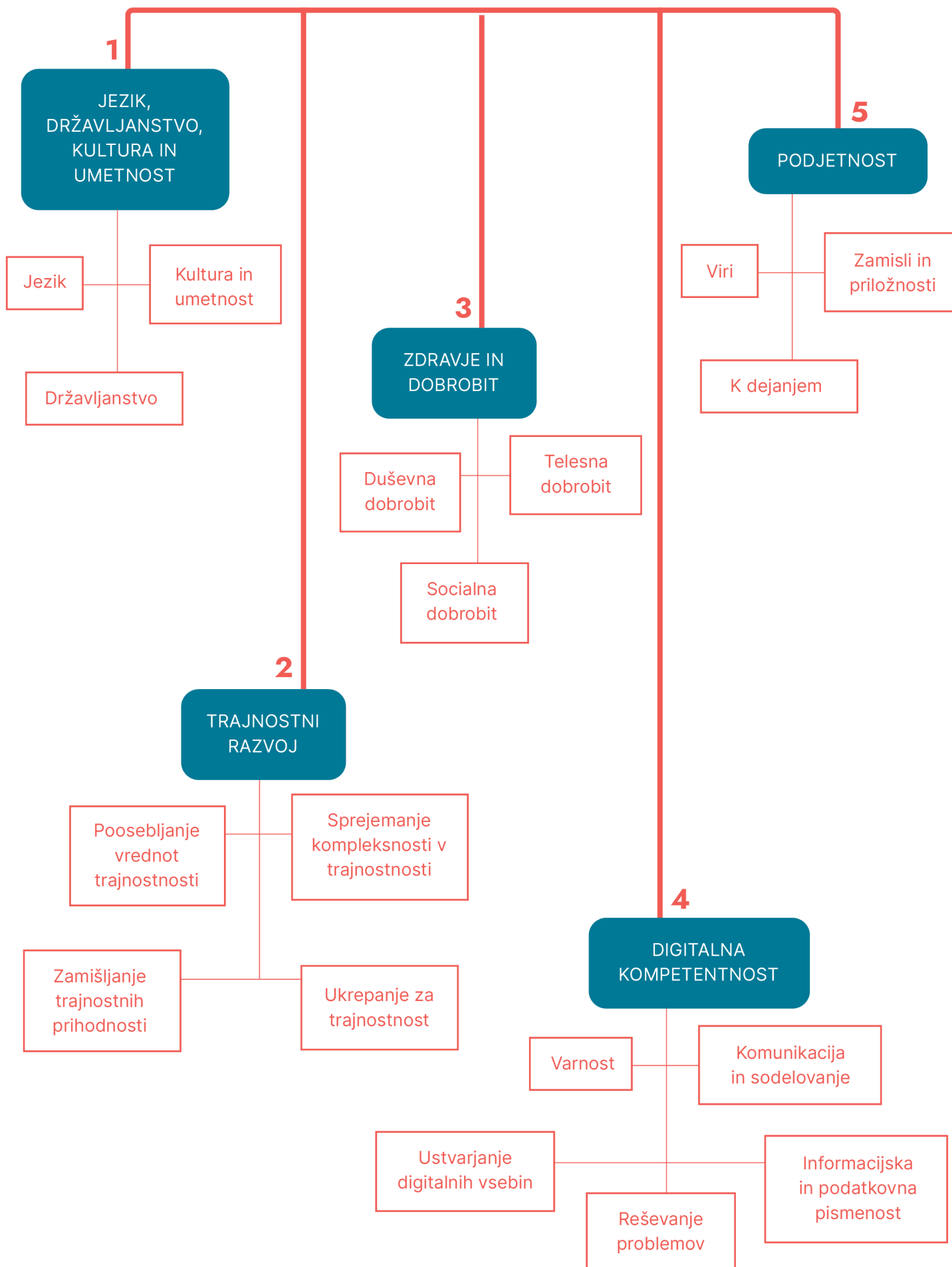
Pri načrtovanju dejavnosti za razvijanje kompetenc 4.1 Skrb za varnost naprav (<https://aplikacijaun.zrss.si/api/link/hy2l3ly>) in 4.2 Varovanje osebnih podatkov in zasebnosti (<https://aplikacijaun.zrss.si/api/link/2byqenc>) učitelju lahko pomaga okvir DigComp 2.2 z opisi znanj, spretnosti in stališč za posamezno kompetenco. Pri tem načrtuje tudi dejavnosti za dijake, kjer razpravljajo o državljski etiki in vrednotah: spoštovanje, človekovo dostojanstvo, pravičnost, resnica, zakon, nediskriminacija in strpnost, ter s tem razvijajo skupni cilj  (1.2.1.5) s področja Jezik, državljanstvo, kultura in umetnost.

(Osebn) podatki in UI: To je zlasti pomembno v luči predpisov GDPR. Hkrati je dijake treba opozoriti na možnost zavarovanja s strani generativnih jezikovnih modelov.

PRILOGE



KLJUČNI CILJI PO PODROČJIH SKUPNIH CILJEV



1 /// JEZIK, DRŽAVLJANSTVO, KULTURA IN UMETNOST



SKUPNI CILJI IN NJIHOVO UMEŠČANJE V UČNE NAČRTE TER KATALOGE ZNANJ

POVZETO PO PUBLIKACIJI WWW.ZRSS.SI/PDF/SKUPNI_CILJI.PDF

1.1 JEZIK

1.1.1 Strokovna besedila	1.1.1.1 Pri posameznih predmetih razvija zmožnost izražanja v različnih besedilnih vrstah (referat, plakat, povzetek, opis, pogovor itd.).
1.1.2 Strokovni jezik	1.1.2.1 Se zaveda, da je učenje vsebine posameznega predmeta hkrati tudi spoznavanje njegove strokovne terminologije; torej učenje jezika na ravni poimenovanj za posamezne pojme in na ravni logičnih povezav. 1.1.2.2 Se izraža z ustrezno terminologijo predmeta in skrbi za ustrezno govorno ter pisno raven svojega strokovnega jezika.
1.1.3 Univerzalni opis jezika kot sistema	(pri vseh predmetih:) 1.1.3.1 Se zaveda podobnosti ter razlik med jeziki in je na to pozoren tudi pri uporabi gradiv v tujih jezikih, pri uporabi prevajalnikov, velikih jezikovnih modelov, avtomatsko prevedenih spletnih strani itd. (pri vseh jezikovnih predmetih:) 1.1.3.2 Se zaveda, da različne jezike lahko opisujemo na podoben način; pri pouku tujih jezikov zato uporablja znanje, pridobljeno pri pouku učnega jezika, in obratno: zna primerjati jezike, ki se jih uči, in razpozna podobnosti ter razlike med njimi.
1.1.4 Razumevanje pomena branja	1.1.4.1 Pri vseh predmetih redno bere, izbira raznolika bralna gradiva, jih razume, poglobljeno analizira in kritično vrednoti.
1.1.5 Jezik in nenasilna komunikacija	1.1.5.1 Razvija lastne sporazumevalne zmožnosti skozi nenasilno komunikacijo.

1.2 DRŽAVLJANSTVO

1.2.1 Poznavanje in privzemanje človekovih pravic ter dolžnosti kot temeljnih vrednot in osnov državljanske etike	1.2.1.1 Pozna, razume in sprejema človekove pravice kot skupni evropski in ustavno določeni okvir skupnih vrednot ter etike. 1.2.1.2 Razume, da so človekove pravice univerzalne in nepogojene, da uveljavljajo vrednote svobode in enakosti. 1.2.1.3 Razume in sprejema, da je obstoj pravic pogojen s spoštovanjem individualne dolžnosti do enake pravice drugega. 1.2.1.4 Razume in sprejema, da je dolžnost do enake pravice drugega dolžnost zaradi dolžnosti (ne le pravica zaradi individualnega interesa). 1.2.1.5 Razume in sprejema človekove pravice in dolžnosti kot osnovno, vsem državljanom skupno etiko (moralo), ki uveljavlja vrednote spoštovanja človekovega dostojanstva, pravičnosti, resnice, zakona, lastnine, nediskriminacije in strpnosti.
1.2.2 Etična refleksija	1.2.2.1 Spozna, da obstajajo moralna vprašanja, pri katerih ni vnaprej danih od vseh sprejetih odgovorov. 1.2.2.2 Razvija občutljivost za moralna vprašanja ter sposobnost, da o njih razmišlja skupaj z drugimi.
1.2.3 Sodelovanje z drugimi v skupnosti in za skupnost	1.2.3.1 Z namenom uresničevanja skupnega dobrega sodeluje z drugimi ter podaja in uresničuje predloge, ki kakovostno spreminjajo skupnosti. 1.2.3.2 Prek lastnega delovanja ozavešča pomen skrbi za demokratično skupnost ter krepi zavedanje o pomenu pripadnosti skupnosti za lastnodobrobit in dobrobit drugih.
1.2.4 Aktivno državljanstvo in politična angažiranost	1.2.4.1 Ob zavedanju pozitivnega pomena politike kot skupnega reševanja izzivov in skrbi za dobrobit vseh, pa tudi iskanja kompromisov in preseganja konfliktov, pozna raznolike oblike demokratičnega političnega angažiranja in se vključuje v politične procese, ki vplivajo na življenja ljudi.
1.2.5 Znanje za kritično mišljenje, za aktivno državljansko držo	1.2.5.1 Uporabi znanja vsakega predmetnega področja za kritično in aktivno državljansko držo.

1.3 KULTURA IN UMETNOST

1.3.1 Sprejemanje, doživljanje in vrednotenje kulture in umetnosti	1.3.1.1 Intuitivno ali zavestno (individualno in v skupini) vzpostavlja odnos do kulture, umetnosti, umetniške izkušnje in procesov ustvarjanja ter ob tem prepozna lastna doživetja in se vživlja v izkušnjo drugega. itd.).
1.3.2 Raziskovanje in spoznavanje kulture ter umetnosti	1.3.2.1 Raziskuje in spozna kulturo, umetnost, umetniške zvrsti ter njihova izrazna sredstva v zgodovinskem in kulturnem kontekstu.
1.3.3 Izražanje v umetnosti in z umetnostjo	1.3.3.1 Je radoveden in raziskuje materiale in umetniške jezike, se z njimi izraža, razvija domišljijo ter pogloblja in širi znanje tudi na neumetniških področjih.
1.3.4 Uživanje v ustvarjalnem procesu ter dosežkih kulture in umetnosti	1.3.4.1 Uživa v ustvarjalnosti, se veseli lastnih dosežkov in dosežkov drugih. 1.3.4.2 V varnem, odprtem in spodbudnem učnem okolju svobodno izraža želje in udejanja ustvarjalne ideje.
1.3.5 Živi kulturo in umetnost	1.3.5.1 Živi kulturo in umetnost kot vrednoto v domačem in šolskem okolju ter prispeva k razvoju šole kot kulturnega središča in njenemu povezovanju s kulturnim in z družbenim okoljem.

2 /// TRAJNOSTNI RAZVOJ



SKUPNI CILJI IN NJIHOVO UMEŠČANJE V UČNE NAČRTE TER KATALOGE ZNANJ

POVZETO PO PUBLIKACIJI WWW.ZRSS.SI/PDF/SKUPNI_CILJI.PDF

2.1 POOSEBLJANJE VREDNOT TRAJNOSTNOSTI

2.1.1 Vrednotenje trajnostnosti	2.1.1.1 Kritično oceni povezanost lastnih vrednot in vrednot družbe s trajnostnostjo glede na svoje trenutne zmožnosti ter družben položaj.
2.1.2 Podpiranje pravičnosti	2.1.2.1 Pri svojem delovanju upošteva etična načela pravičnosti, enakopravnosti ter sočutja.
2.1.3 Promoviranje narave	2.1.3.1 Odgovoren odnos do naravnih sistemov gradi na razumevanju njihove kompleksnosti in razmerij med naravnimi ter družbenimi sistemi.

2.2 SPREJEMANJE KOMPLEKSNOSTI V TRAJNOSTNOSTI

2.2.1 Sistemsko mišljenje	2.2.1.1 K izbranemu problemu pristopa celostno, pri čemer upošteva povezanost okoljskega, gospodarskega in družbenega vidika. 2.2.1.2 Presoja kratkoročne in dolgoročne vplive delovanja posameznika in družbenih skupin v družbi, družbe na lokalni, regionalni, nacionalni ter globalni ravni.
2.2.2 Kritično mišljenje	2.2.2.1 Kritično presoja informacije, poglede in potrebe o trajnostnem razvoju z vidika naravnega okolja, živih bitij in družbe, pri čemer upošteva različne poglede, pogoje z osebnim, socialnim in kulturnim ozadjem.
2.2.3 Formuliranje problema	2.2.3.1 Pri opredelitvi problema upošteva značilnosti problema – (ne)jasnost, (ne)opredeljenost, (ne)določljivost problema – in lastnosti reševanja – (ne)definirane, (ne)sistemske rešitve – ter vpletenost deležnikov.

2.3 ZAMIŠLJANJE TRAJNOSTNIH PRIHODNOSTI

2.3.1 Pismenost za prihodnost	2.3.1.1 Na podlagi znanja, znanstvenih dognanj in vrednot trajnostnosti razume ter vrednoti možne, verjetne in želene trajnostne prihodnosti (scenarije). 2.3.1.2 Presoja dejanja, ki so potrebna za doseganje želene trajnostne prihodnosti.
2.3.2 Prilagodljivost	2.3.2.1 V prizadevanju za trajnostno prihodnost tvega in se kljub negotovostim prilagaja ter sprejema trajnostne odločitve za svoja dejanja.
2.3.3 Raziskovalno mišljenje	2.3.3.1 Pri načrtovanju in reševanju kompleksnih problemov/-atik trajnostnosti uporablja in povezuje znanja in metode različnih znanstvenih disciplin ter predlaga ustvarjalne in inovativne ideje in rešitve.

2.4 UKREPANJE ZA TRAJNOSTNOST

2.4.1 Politična angažiranost	2.4.1.1 Ob upoštevanju demokratičnih načel kritično vrednoti politike z vidika trajnostnosti ter sodeluje pri oblikovanju trajnostnih politik in prakse na lokalni, regionalni, nacionalni ter globalni ravni.
2.4.2 Kolektivno ukrepanje	2.4.2.1 Pri prizadevanju in ukrepanju za trajnostnost upošteva demokratična načela in aktivno ter angažirano (konstruktivno) sodeluje z drugimi.
2.4.3 Individualna iniciativa	2.4.3.1 Se zaveda lastnega potenciala in odgovornosti za trajnostno delovanje in ukrepanje na individualni, kolektivni ter politični ravni.



3.1 DUŠEVNA DOBROBIT

3.1.1 Samozavedanje	3.1.1.1 Zaznava in prepoznavna lastno doživljanje (telesne občutke, čustva, misli, vrednote, potrebe, želje) ter lastno vedenje. 3.1.1.2 Skozi izkušnje razume povezanost različnih vidikov lastnega doživljanja (telesni občutki, čustva, misli, vrednote, potrebe, želje) v specifičnih situacijah (npr. ob doživljanju uspeha in neuspeha v učnih in socialnih situacijah).
3.1.2 Samouravnavanje	3.1.2.1 Uravnava lastno doživljanje in vedenje (npr. v stresnih učnih in socialnih situacijah). 3.1.2.2 Razvija samozaupanje in samospoštovanje. 3.1.2.3 Se učinkovito spoprijema z negotovostjo in kompleksnostjo. 3.1.2.4 Učinkovito upravlja, organizira čas učenja in prosti čas.
3.1.3 Postavljanje ciljev	3.1.3.1 Prepoznavna lastne interese, lastnosti, močna in šibka področja ter v skladu z njimi načrtuje kratkoročne in dolgoročne cilje (vezane na duševno, telesno, socialno in učno področje). 3.1.3.2 Spremlja doseganje in spreminjanje načrtovanih ciljev. 3.1.3.3 Razvija zavzetost in vztrajnost ter krepi zmožnosti odložitve nagrade.
3.1.4 Prožen način razmišljanja	3.1.4.1 Razvija osebno prožnost, razvojno miselno naravnost, radovednost, optimizem in ustvarjalnost. 3.1.4.2 Učinkovito se spoprijema s problemskimi situacijami, ki zahtevajo proaktivno miselno naravnost.
3.1.5 Odgovornost in avtonomija	3.1.5.1 Krepi odgovornost, avtonomijo in skrbi za osebno integriteto.

3.2 TELESNA DOBROBIT

3.2.1 Gibanje in sedenje	3.2.1.1 Razume pomen vsakodnevnega gibanja za zdravje in dobro počutje. 3.2.1.2 Razvija pozitiven odnos do gibanja. 3.2.1.3 Vključuje se v različne gibalne dejavnosti, tudi tiste, ki razbremenijo naporen vsakdan. 3.2.1.4 Razume škodljivosti dolgotrajnega sedenja, razvija navade za prekinitev in zmanjšanje sedenja.
3.2.2 Prehrana in prehranjevanje	3.2.2.1 Razume pomen uravnotežene prehrane. 3.2.2.2 Razvija navade zdravega prehranjevanja. 3.2.2.3 Oblikuje pozitiven odnos do hrane in prehranjevanja.
3.2.3 Sprostitev in počitek	3.2.3.1 Razume pomen počitka in sprostitve po miselnem ali telesnem naporu. 3.2.3.2 Spozna različne tehnike sproščanja in uporablja tiste, ki mu najbolj koristijo. 3.2.3.3 Razume pomen dobrih spalnih navad za učinkovito telesno in miselno delovanje ter ravnanje.
3.2.4 Varnost	3.2.4.1 Spozna različne zaščitne ukrepe za ohranjanje zdravja. 3.2.4.2 Ravna varno in odgovorno, pri čemer skrbi za ohranjanje zdravja sebe in drugih.
3.2.5 Preventiva pred različnimi oblikami zasvojenosti	3.2.5.1 Usvaja znanje o oblikah in stopnjah zasvojenosti ter o strategijah, kako se jim izogniti oz. jih preprečiti z zdravim življenjskim slogom.

3.3 SOCIALNA DOBROBIT

3.3.1 Socialno zavedanje in zavedanje raznolikosti	3.3.1.1 Ozavešča lastno doživljanje in vedenje v odnosih z drugimi. 3.3.1.2 Zaveda se in prepoznavna raznolikosti v ožjem (oddelek, vrstniki, družina) in širšem okolju (šola, lokalna skupnost, družba).
3.3.2 Komunikacijske spretnosti	3.3.2.1 Razvija spretnosti aktivnega poslušanja, asertivne komunikacije, izražanja interesa in skrbi za druge.
3.3.3 Sodelovanje in reševanje konfliktov	3.3.3.1 Konstruktivno sodeluje v različnih vrstah odnosov na različnih področjih. 3.3.3.2 Krepi sodelovalne veščine, spretnosti vzpostavljanja in vzdrževanja kakovostnih odnosov v oddelku, družini, šoli, širši skupnosti, temelječih na spoštljivem in asertivnem komuniciranju ter enakopravnosti.
3.3.4 Empatija	3.3.4.1 Krepi zmožnosti razumevanja in prevzemanja perspektive drugega. 3.3.4.2 Prepoznavna meje med doživljanjem sebe in drugega. 3.3.4.3 Uravnava lastno vedenje v odnosih, upoštevajoč več perspektiv hkrati.
3.3.5 Prosocialno vedenje	3.3.5.1 Krepi prepoznavanje in zavedanje potrebe po nudenju pomoči drugim. 3.3.5.2 Prepoznavna lastne potrebe po pomoči in strategije iskanja pomoči. 3.3.5.3 Krepi družbeno odgovornost.

4 /// DIGITALNA KOMPETENTNOST



SKUPNI CILJI IN NJIHOVO UMEŠČANJE V UČNE NAČRTE TER KATALOGE ZNANJ

POVZETO PO PUBLIKACIJI WWW.ZRSS.SI/PDF/SKUPNI_CILJI.PDF

4.1 INFORMACIJSKA IN PODATKOVNA PISMENOST

4.1.1 Brskanje, iskanje in filtriranje podatkov, informacij ter digitalnih vsebin	4.1.1.1 Izraža informacijske potrebe, išče podatke, informacije in vsebine v digitalnih okoljih ter izboljšuje osebne strategije iskanja.
4.1.2 Vrednotenje podatkov, informacij in digitalnih vsebin	4.1.2.1 Analizira, primerja in kritično vrednoti verodostojnost in zanesljivost podatkov, informacij in digitalnih vsebin.
4.1.3 Upravljanje podatkov, informacij in digitalnih vsebin	4.1.3.1 Podatke zbira, obdeluje, prikazuje in shranjuje na najustreznejša mesta (trdi disk, oblak, USB itd.) tako, da jih zna kasneje tudi najti.

4.2 KOMUNIKACIJA IN SODELOVANJE

4.2.1 Interakcija z uporabo digitalnih tehnologij	4.2.1.1 Sporazumeva se z uporabo različnih digitalnih tehnologij in razume ustrezna sredstva komunikacije v danih okoliščinah.
4.2.2 Deljenje z uporabo digitalnih tehnologij	4.2.2.1 Deli podatke, informacije in digitalne vsebine z drugimi z uporabo ustreznih digitalnih tehnologij. Deluje v vlogi posrednika in je seznanjen s praksami navajanja virov ter avtorstva.
4.2.3 Državljsko udejstvovanje z uporabo digitalnih tehnologij	4.2.3.1 Išče in uporablja portale za udejstvovanje v družbi. Poišče skupine, ki zastopajo njegove interese, s pomočjo katerih lahko aktivno daje predloge za spremembe.
4.2.4 Sodelovanje z uporabo digitalnih tehnologij	4.2.4.1 Z digitalnimi orodji soustvarja skupno vsebino in se zavzema za krepitev sodelovanja med člani.
4.2.5 Spletni bonton	4.2.5.1 Pri uporabi digitalnih tehnologij in omrežij prilagaja svoje vedenje pričakovanjem ter pravilom, ki veljajo v določeni skupini.
4.2.6 Upravljanje digitalne identitete	4.2.6.1 Ustvari eno ali več digitalnih identitet in z njimi upravlja, skrbi za varovanje lastnega ugleda ter za ravnanje s podatki, ki nastanejo z uporabo številnih digitalnih orodij in storitev v različnih digitalnih okoljih.

4.3 USTVARJANJE DIGITALNIH VSEBIN

4.3.1 Razvoj digitalnih vsebin	4.3.1.1 Ustvarja in ureja digitalne vsebine v različnih formatih.
4.3.2 Umeščanje in poustvarjanje digitalnih vsebin	4.3.2.1 Digitalne vsebine obogati z dodajanjem slik, glasbe, videoposnetkov, vizualnih efektov idr.
4.3.3 Avtorske pravice in licence	4.3.3.1 Pozna licence in avtorske pravice. Spoštuje pravice avtorjev in jih ustrezno citira.
4.3.4 Programiranje	4.3.4.1 Zna ustvariti algoritem za rešitev enostavnega problema. Zna odpraviti preproste napake v programu.

4.4 VARNOST

4.4.1 Varovanje naprav	4.4.1.1 Deluje samozaščitno; svoje naprave zaščiti z ustreznimi gesli, programsko opremo in varnim ravnanjem, jih ne pušča brez nadzora v javnih prostorih idr.
4.4.2 Varovanje osebnih podatkov in zasebnosti	4.4.2.1 Varuje osebne podatke in podatke drugih ter prepoznava zaupanja vredne ponudnike digitalnih storitev.
4.4.3 Varovanje zdravja in dobrobiti	4.4.3.1 Digitalno tehnologijo uporablja uravnoteženo, skrbi za dobro telesno in duševno dobrobit ter se izogiba negativnim vplivom digitalnih medijev.
4.4.4 Varovanje okolja	4.4.4.1 Zaveda se vplivov digitalnih tehnologij in njihove uporabe na okolje.

4.5 REŠEVANJE PROBLEMOV

4.5.1 Reševanje tehničnih težav	4.5.1.1 Prepozna tehnične težave pri delu z napravami ali digitalnimi okolji ter jih rešuje.
4.5.2 Prepoznavanje potreb in opredelitev tehnoloških odzivov	4.5.2.1 Prepoznava in ocenjuje potrebe, izbira ter uporablja digitalna orodja in jih prilagaja lastnim potrebam.
4.5.3 Ustvarjalna uporaba digitalne tehnologije	4.5.3.1 S pomočjo digitalne tehnologije ustvarja rešitve in novosti v postopkih ter izdelkih.
4.5.4 Prepoznavanje vrzeli v digitalnih kompetencah	4.5.4.1 Prepozna vrzeli v svojih digitalnih kompetencah, jih po potrebi izboljšuje in dopolnjuje ter pri tem podpira tudi druge.



5.1 ZAMISLI IN PRILOŽNOSTI

5.1.1 Odkrivanje priložnosti	5.1.1.1 Prepoznavava avtentične izzive kot priložnosti za ustvarjanje vrednost zase in za druge.
5.1.2 Ustvarjalnost in inovativnost	5.1.2.1 Pri reševanju izzivov na ustvarjalen način uporablja znanje in izkušnje za ustvarjanju boljših rešitev. 5.1.2.2 Pripravi nabor možnih rešitev izziva, pri čemer stremi k oblikovanju vrednosti za druge.
5.1.3 Vizija	5.1.3.1 Oblikuje vizijo prihodnosti, ki vključuje odgovore na vprašanja, kaj namerava početi v prihodnosti, kakšen želi postati in kakšno skupnost želi sooblikovati.
5.1.4 Vrednotenje zamisli	5.1.4.1 Vrednoti rešitve ob upoštevanju kriterijev po načelu ustvarjanja dobrobiti za druge in izbere ustrezno rešitev.
5.1.5 Etično in trajnostno razmišljanje	5.1.5.1 Prepozna in ovrednoti vpliv svojih odločitev ter ravnanj na skupnost in okolje.

5.2 VIRI

5.2.1 Samozavedanje in samoučinkovitost	5.2.1.1 Prepozna svoje želje, močna in šibka področja ter zaupa, da lahko pozitivno vpliva na ljudi in situacije.
5.2.2 Motiviranost in vztrajnost	5.2.2.1 Je pozitivno naravnani, samozavesten in osredotočen na proces reševanja izziva. 5.2.2.2 Vztraja pri opravljanju kompleksnejših nalog.
5.2.3 Vključevanje virov	5.2.3.1 Pridobi podatke in sredstva (materialna, nematerialna in digitalna), potrebne za prehod od zamisli k dejanjem, s katerimi odgovorno in učinkovito upravlja, pri čemer upošteva učinkovito izrabo lastnega časa in finančnih sredstev.
5.2.4 Vključevanje človeških virov	5.2.4.1 Spodbuja in motivira druge pri reševanju skupnih nalog. 5.2.4.2 Poišče ustrezno pomoč posameznika (npr. vrstnika, učitelja, strokovnjak itd.) ali strokovne skupnosti. 5.2.4.3 Razvije spretnosti za učinkovito komunikacijo, pogajanje, vodenje, ki so potrebne za doseganje rezultatov.
5.2.5 Finančna pismenost	5.2.5.1 V različnih življenjskih situacijah prepozna in rešuje finančne izzive (viri finančnih sredstev, skrb za denar, poslovanje npr. z banko, zavarovalnico ter drugimi finančnimi institucijami, ocena potrebnih sredstev, tveganj in odločitev o zadolževanju ter naložbi, varčevanju). 5.2.5.2 Sprejema odgovorne finančne odločitve za doseganje blaginje (osebne in za skupnost). 5.2.5.3 Pridobiva ustrezno znanje na področju finančne pismenosti za kakovostno vsakdanje in poklicno življenje.

5.3 K DEJANJEM

5.3.1 Prevzemanje pobude	5.3.1.1 Spodbuja in sodeluje pri reševanju izzivov, pri čemer prevzema individualno in skupinsko odgovornost.
5.3.2 Načrtovanje in upravljanje	5.3.2.1 V procesu reševanja problemov si zastavlja kratkoročne, srednjeročne in dolgoročne cilje, opredeli prednostne naloge in pripravi načrt. 5.3.2.2 Prilagaja se nepredvidnim spremembam.
5.3.3 Obvladovanje negotovosti, dvoumnosti in tveganja	5.3.3.1 Na nepredvidene situacije se odziva s pozitivno naravnanoostjo in ciljem, da jih uspešno razreši. 5.3.3.2 Pri odločitvah primerja in analizira različne informacije, da zmanjša negotovost, dvoumnost in tveganja.
5.3.4 Sodelovanje	5.3.4.1 Sodeluje z različnimi posamezniki ali skupinami. 5.3.4.2 Morebitne spore rešuje na konstruktiven način in po potrebi sklepa kompromise.
5.3.5 Izkusveno učenje	5.3.5.1 Z reševanjem izzivov pridobiva nove izkušnje in jih upošteva pri sprejemanju nadaljnjih odločitev. 5.3.5.2 Presoja uspešnost doseganja zastavljenih ciljev in pri tem prepozna priložnosti za nadaljnje učenje. 5.3.5.3 Prepozna priložnosti za aktivno uporabo pridobljenega znanja v novih situacijah. 5.3.5.4 Povratne informacije uporabi za nadaljnji razvoj podjetnostne kompetence.