

LV1 – FUNKCIJE: OD PROBLEMA DO PROGRAMSKOG RJEŠENJA

Naziv modula	Programiranje
Skup ishoda učenja	SIU1 – Funkcije i rekurzija u programiranju
Ishod SIU-a	Odrediti ulazne i izlazne parametre funkcionalnih cjelina.
Naziv laboratorijske vježbe (UTR)	LV1 – Funkcije: od problema do programskog rješenja
Trajanje	2 školska sata
Ime i prezime / datum	_____

ISHODI VJEŽBE

- analizirati problemski zadatak i odrediti potrebne ulazne podatke i očekivani rezultat
- izdvojiti postupak koji je smisljeno oblikovati kao funkciju
- odrediti parametre i povratnu vrijednost funkcije
- zapisati definiciju funkcije i pravilno je pozvati
- povezati više poziva iste funkcije s različitim argumentima
- izraditi i testirati jednostavan program koji koristi funkcije
- osmisliti testne primjere, uključujući rubne slučajeve
- objasniti vlastito programsko rješenje i obrazložiti izbor parametara i rezultata funkcije

PRIPREMA

Prije programiranja u bilježnicu kratko odgovorite:

1. Što predstavlja ulaz, a što izlaz funkcije?
2. Koja je razlika između definicije funkcije i njezina poziva?
3. Napišite primjer poziva funkcije koja prima dva cijela broja i vraća veći od njih.

Napomena: Priprema služi aktiviranju prethodnog znanja i ne treba trajati dulje od 5–7 minuta.

OPĆE UPUTE ZA RAD

Za svaki problemski zadatak primjenjujte isti postupak:

RAZUMIJ PROBLEM → ODREDI ULAZ I IZLAZ → OSMISLI FUNKCIJU → PREDVIDI REZULTAT → NAPIŠI PROGRAM → TESTIRAJ → PRONAĐI SLABU TOČKU → DORADI

- Prije pisanja programskog kôda najprije odredite što funkcija treba primiti i što treba vratiti.
- Za svaki obvezni problem osmislite najmanje jedan vlastiti testni primjer i zapišite očekivani rezultat prije pokretanja programa.
- Važnije je razumjeti vlastito rješenje nego imati dovršen program koji ne možete objasniti.
- Ako radite u paru, oba učenika moraju razumjeti cijelo rješenje. Tijekom rada zamijenite uloge: jedan učenik radi za tipkovnicom, drugi prati algoritam, provjerava i priprema testove, a zatim zamijenite uloge.
- Ako radite u paru, svaki učenik osmišljava svoj testni primjer i mora moći objasniti što njime provjerava.
- Nastavnik može tijekom rada ili pri pregledu zatražiti da bilo koji učenik objasni dio programa, predvidi rezultat dodatnog testa ili samostalno ispravi/dopuni dio rješenja.
- Vježba se ne ocjenjuje. Služi učenju, uvježbavanju i pripremi za kasnije vrednovanje ishoda.

UPORABA GENERATIVNOG AI-JA

Generativni AI smije se koristiti kao pomoć pri učenju: za objašnjenje pogreške, prijedlog testnog primjera, provjeru ideje ili raspravu o mogućem poboljšanju rješenja. Ne treba ga koristiti kao zamjenu za vlastito rješavanje problema. Ako je AI značajnije korišten, učenik navodi za što ga je koristio i zapisuje prompt. Svaki dio programskog kôda koji se predaje učenik mora razumjeti, moći objasniti, izmijeniti i po potrebi ispraviti.

Preporučeni način uporabe AI-ja pri testiranju: zatražite test koji bi mogao otkriti pogrešku u vašem programu, ali sami odredite očekivani rezultat prije pokretanja.

I. PROBLEMSKI ZADATAK – TURNIR

Na programerskom turniru natjecatelj rješava tri zadatka. Za svaki zadatak može osvojiti od 0 do 100 bodova. Natjecatelj prolazi u sljedeći krug ako ukupno ima najmanje 200 bodova i iz svakog pojedinačnog zadatka ima najmanje 40 bodova.

Napišite program koji za rezultate više natjecatelja određuje prolaze li u sljedeći krug.

Prije programiranja

Naziv funkcije: _____

Ulazni parametri: _____

Povratna vrijednost: _____

Što radi main(), a što funkcija? _____

Zahtjevi

- Oblikujte odgovarajuću funkciju.
- Istu funkciju pozovite više puta s različitim podacima.
- Program mora jasno ispisati prolazi li natjecatelj u sljedeći krug.

Zadani testovi

Bodovi	Očekivani rezultat
75 82 51	PROLAZI
100 100 35	NE PROLAZI
60 70 70	PROLAZI
40 40 100	NE PROLAZI

Vlastiti test

Ulaz: _____

Očekivani izlaz: _____

Što ovim testom provjeravam: _____

Mala dorada

Organizator mijenja pravilo: potrebno je najmanje 180 bodova ukupno, dok ostali uvjeti ostaju isti. Prilagodite program i odgovorite: koliko ste mjesta u programu morali promijeniti i zašto?

II. PROBLEMSKI ZADATAK – SIGURNA ZONA

U računalnoj igri igrač se kreće po koordinatnoj mreži. Sigurna zona je pravokutnik određen koordinatama x_{min} , x_{max} , y_{min} i y_{max} . Za zadanu poziciju igrača (x, y) treba odrediti nalazi li se igrač unutar sigurne zone ili na njezinu rubu.

Za zonu $x_{min} = 2$, $x_{max} = 8$, $y_{min} = 3$, $y_{max} = 7$ razmotrite točke $(5,5)$, $(2,7)$, $(9,5)$ i $(2,2)$.

1. Riješite problem bez programiranja

Za svaku zadanu točku odredite očekivani rezultat i obrazložite barem jedan granični slučaj.

2. Projektirajte funkciju

Što funkcija mora primiti? _____

Što mora vratiti? _____

Kako biste je nazvali? _____

Tek nakon toga napišite funkciju i program koji je koristi.

3. Testirajte rješenje

- jedan test za točku unutar zone
- jedan test za točku na rubu
- jedan test za točku izvan zone

Za svaki test predvidite rezultat prije pokretanja programa.

4. Napadni vlastiti program – AI kao tester

Ako želite, zatražite od AI alata da predloži jedan testni primjer kojim bi pokušao otkriti pogrešku u vašem programu. AI ne treba pisati rješenje ni izračunati očekivani rezultat.

Primjer prompta: „Nemoj mi rješavati zadatak niti pisati programski kod. Predloži jedan testni primjer kojim bi pokušao otkriti pogrešku u programu koji provjerava nalazi li se točka unutar pravokutne zone. Objasni samo što tim primjerom pokušavaš provjeriti.”

AI test: _____

Zašto je zanimljiv: _____

Moj očekivani rezultat: _____

Je li otkrio pogrešku? Ako jest, koju? _____

III. IZAZOV – NAJBLIŽI CILJU

Izazov nije obvezan. Namijenjen je svima koji nakon obveznih zadataka žele pokušati zahtjevniji problem.

U igri se igrač nalazi na koordinati x . Na pravcu se nalaze dva cilja na koordinatama a i b . Potrebno je odrediti koji je cilj bliži igraču. Ako su oba cilja jednako udaljena, treba ispisati JEDNAKO.

Primjer: $x = 10$, $a = 4$, $b = 13 \rightarrow$ bliži je cilj B.

Prije programiranja razmislite: je li za ovaj problem prikladnija jedna funkcija ili više njih? Obrazložite svoj izbor.

Ulaz	Očekivani rezultat
10 4 13	B
10 7 15	A
10 5 15	JEDNAKO

Osmislite najmanje jedan vlastiti test koji smatrate posebno dobrim i obrazložite zašto.

ZAVRŠETAK I PREDAJA

- Do završetka UTR sata predajte trenutno stanje svojih programa, čak i ako niste riješili sve zadatke.
- Za svaki obvezni problem predaje se programska datoteka (.cpp).
- Na početku programske datoteke u komentaru navedite ime učenika (ili oba učenika ako se radilo u paru), što funkcija prima i vraća, vlastiti test s očekivanim rezultatom te kratku bilješku o uporabi AI-ja ako je korišten.
- Ako se radi u paru, svaki učenik mora imati svoj testni primjer, a oba učenika moraju moći objasniti cijeli program.
- Nastavnik može zatražiti kratko objašnjenje, dodatni test ili samostalnu dopunu/ispravak programa.
- Vježba se ne ocjenjuje. Važnije je predati vlastito nedovršeno rješenje koje možete objasniti nego dovršeno rješenje koje ne razumijete. Nije potreban zaseban Word/PDF izvještaj.

Predložak komentara na početku .cpp datoteke

```
/*  
LV1 – Zadatak: _____  
Učenik/učenici: _____  
  
Funkcija prima: _____  
Funkcija vraća: _____  
  
Moj test:  
ulaz: _____  
očekivani rezultat: _____  
što test provjerava: _____  
  
AI: nisam koristio / koristio za _____  
Prompt (ako je korišten): _____  
*/
```

ORGANIZACIJSKA NAPOMENA

Vježba je oblikovana tako da se može provoditi i kada raspored sati nije idealan. Problemski zadatak A može se raditi već nakon prvog sata VPUiP-a. Problemski zadatak B i Izazov prirodno se nastavlja nakon drugog sata VPUiP-a. Ako su oba VPUiP sata održana prije UTR-a, vježba se izvodi redom kao jedinstvena cjelina.

OKVIRNA RASPODJELA VREMENA

Aktivnost	Vrijeme
Priprema i aktivacija prethodnog znanja	5–7 min
Problemski zadatak A – Turnir	25–30 min
Problemski zadatak B – Sigurna zona	30–35 min
Dovršavanje, testiranje, Izazov i kratka analiza	15–20 min