

Priprema za izvođenje nastave – ogledni primjer prilagodbi metoda, sadržaja i aktivnosti za učenike s teškoćama

Škola:	Strukovna škola Đurđevac
Mjesto:	Đurđevac
Obrazovni sektor:	Elektrotehnika i računalstvo
Strukovna kvalifikacija:	Tehničar za računalstvo
Odjel:	1b
Nastavni predmet:	Algoritmi i programiranje
Naziv teme:	Složene naredbe – programske funkcije
Nastavnik:	Tihomir Robotić

Napomena: prilagodba sadržaja, metoda i zahtjeva ovisi o specifičnim teškoćama koje učenik ima. U ovom radu fokus je na spoznajnoj prilagodbi i prilagodbi školskih zahtjeva.

Planirani tijek aktivnosti

Vremensko trajanje aktivnosti	2 školska sata
Odgojno obrazovni ishodi	Odabrati i osmisлити algoritme za rješavanje jednostavnijih problema Primijeniti dekompoziciju programa te dio programa pravilno izdvojiti u funkciju Pozvati funkciju iz glavnog programa Rješenje zapisati dijagramom tijeka
Međupredmetne teme	Učiti kako učiti A.4/5.2. Učenik se koristi različitim strategijama učenja i samostalno ih primjenjuje u ostvarivanju ciljeva učenja i rješavanju problema u svim područjima učenja. B.4/5.4. Učenik samovrednuje proces učenja i svoje rezultate, procjenjuje ostvareni napredak te na temelju toga planira buduće učenje. Osobni i socijalni razvoj A.4.3. Razvija osobne potencijale. A.4.4. Upravlja svojim obrazovnim i profesionalnim putem.
Opis aktivnosti	Uvodni dio Učenik ponavlja pojam dekompozicije i top down metode Očekivano predznanje – raščlanjivanje zadataka na podzadatke, top down metoda Pripremni dio Motivacijsko pitanje – što ćete napraviti s dijelom programskog koda koji vam se u programu više puta ponavlja? Glavni dio – aktivnost 1 Nastavnik koristi metodu demonstracije Demonstriramo deklaraciju i poziv funkcije koja obavlja jednostavan posao, na primjer, zbraja dva broja Zatim demonstriramo isti primjer, ali s različitom funkcijom, na primjer, za množenje Nakon toga demonstriramo primjer u kojem se funkciji prenosi vrijednost, a funkcija obavlja složeniji posao koji uključuje aritmetičke operacije i programske petlje i na kraju, funkcija vrati rezultat obrade Vježbanje – aktivnost 2 U ovoj aktivnosti učenici samostalno rješavaju problem Potrebno je samostalno izraditi program koji od korisnika traži unos broja. Taj broj nam govori koliko brojeva ćemo nakon njega unijeti. Za svaki uneseni broj potrebno je pozvati funkciju prima taj broj kao argument, a kao izlaznu vrijednost ispisuje je li broj paran ili neparan. Program sadrži: Naredbe unosa i ispisa Deklaraciju funkcije i poziv funkcije iz glavnog programa Grananja Programsku petlju s poznatim brojem ponavljanja

	Vrednovanje Učenik dobiva povratnu informaciju o usvojenosti novih sadržaja Nastavnik dobiva povratnu informaciju o stupnju usvojenosti sadržaja od strane učenika Poticanje natjecateljskog duha rješavanjem zadataka i uspoređivanje svojih odgovora s odgovorima drugih učenika Vrednovanje za učenje: Nastavnik prati aktivnosti tijekom rada Učenik kroz razgovor s nastavnikom dobiva povratnu informaciju o svom načinu razmišljanja i tehnikama rješavanja problema Nastavnik kroz odgovore učenika dobiva povratnu informaciju u kojoj mjeri je njegova aktivnost razumljiva te jesu li aktivnosti za učenike razumljivo postavljene Vrednovanje kao učenje: Učenici samovrednuju svoje odgovore Učenici vrednuju odgovore drugih učenika
--	--

Organizacija rada – artikulacija aktivnosti

Nastavna sredstva, pomagala i ostali materijalni uvjeti:

- Informatički praktikum opremljen računalima, projektorom i platnom za prikaz
- Programski alat Flowgorythm

Nastavne metode:

- Usmeno izlaganje
- Razgovor
- Postavljanje pitanja
- Praktičan rad
- Demonstracija
- Samostalno rješavanje zadataka
- Rješavanje problema

Metodički oblici rada:

- Individualni rad
- Frontalni rad

Metode za spoznajno prilagođavanje

- Isticanje bitnoga različitim tipovima obilježavanja i/ili slikovnog predočavanja
- Raščlanjivanje složenih zadataka
- Semantičko pojednostavljivanje sadržaja učenja preradom sadržaja u smislu uporabe jednostavnog govornog izraza

Metode za prilagodbu školskih zahtjeva

- Doziranje zadataka, postepeno uz poticanje samostalnosti u radu

- Mogućnost duljeg vremena za rad

Tijek izvođenja nastave

1. Uvodni dio

Učenik ponavlja pojam dekompozicije i top – down metode

Očekivano predznanje – učenik intuitivno razumije da se složeni problemi rješavaju na način da ih razbijemo u manje cjeline koje nam je lakše riješiti. Učenik je čuo za top – down metodu, ili rješavanje složenih problema po principu „odozgo prema dolje“. Učenik navodi primjere razbijanja složenih problema u jednostavnije iz stvarnog života.

Ukoliko učenik ne može navesti primjere, nastavnik navodi učenika, od učenika se očekuje razrada po koracima, na primjer:

Možete li pojesti kilogram kruha odjednom?	Možete li pročitati knjigu od 500 stranica odjednom?
A ako ga narežete na kriške?	A ako svaki dan pročitate 50?

2. Pripremni dio

Motivacijsko pitanje – što ćete napraviti s dijelom programskog koda koji vam se u programu više puta ponavlja?

Najava glavnog dijela – korištenjem alata Flowgorythm prikazat ćemo rješavanje problema u kojima koristimo petlju s ispitivanjem uvjeta

3. Glavni dio

Aktivnost 1 – Nastavnik koristi metodu demonstracije

Demonstriramo izvođenje naredbe za ponavljanje s ispitivanjem uvjeta na način da zajedno s učenicima izradimo tri primjera u kojima demonstriramo:

- Primjer u kojem se petlja s ispitivanjem uvjeta može iskoristiti za rješavanje zadataka koji zahtijevaju poznat broj ponavljanja
- Primjer u kojem je broj ponavljanja nepredvidiv jer ovisi o korisniku
- Primjer u kojem se petlja s ispitivanjem uvjeta kombinira s aritmetičkim operacijama

Primjer 1 – Krećemo od najjednostavnijeg mogućeg primjera. Zbrajamo dva broja. Pretpostavimo da se operacija zbrajanja može pozvati više puta, uvijek s različitim brojevima. Kako ćete dekomponirati zadatak?

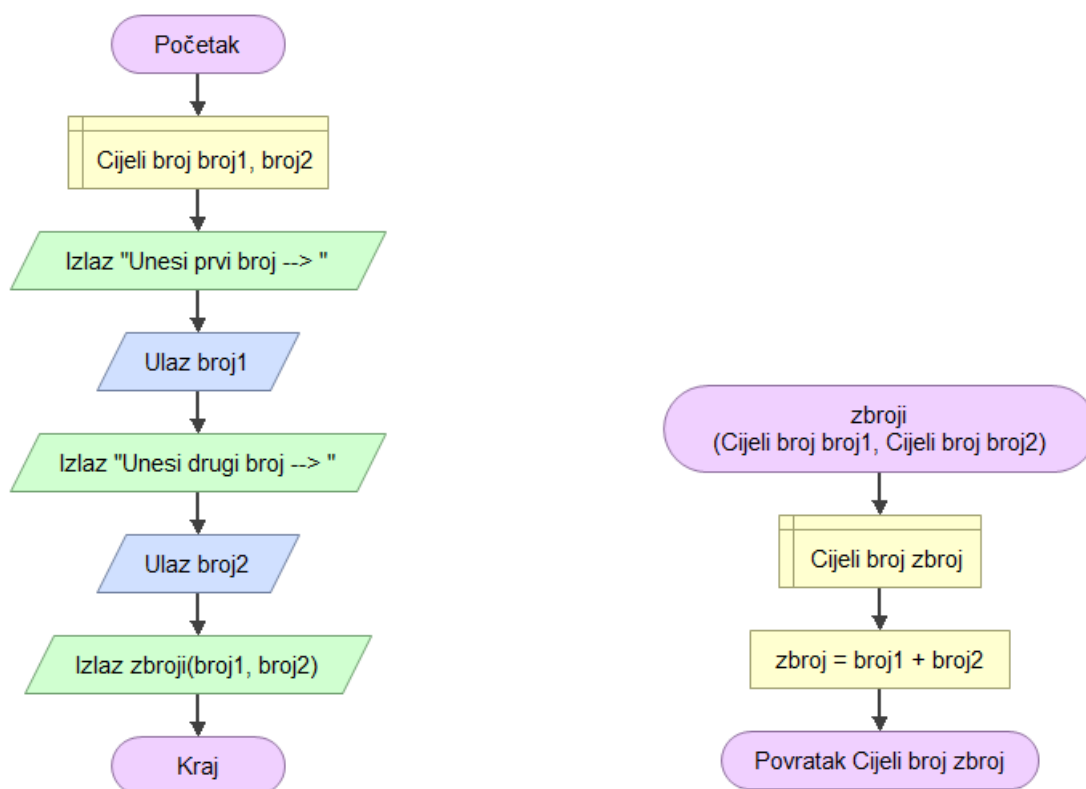
Ulaz

Obrada

Izlaz

Koji dio programa je najlogičnije izdvojiti u posebnu cjelinu?

Nastavnik demonstrira stvaranje funkcije, poziv funkcije iz glavnog programa i ispis rezultat koji funkcija vraća u glavni program



Očekivani izlaz:

Unesi prvi broj →

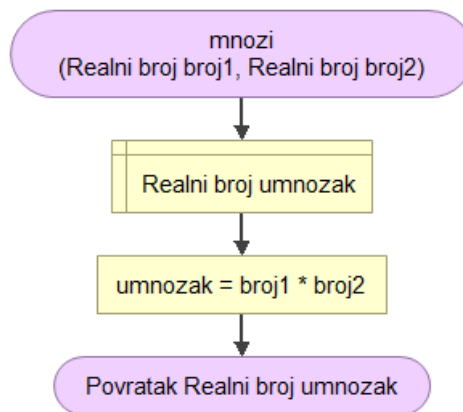
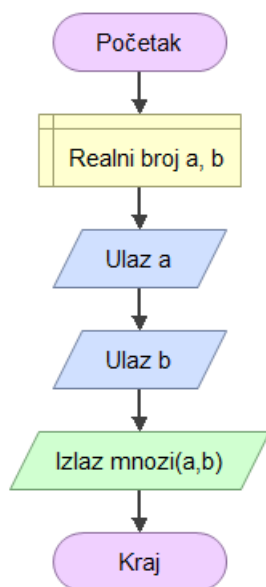
2

Unesi drugi broj →

3

5

Primjer 2 – Pokušajmo u pojednostavljenoj formi ponoviti primjer, ali ovaj put umjesto zbrajanja obavljamo množenje. Ideja je primijetiti da je standardni dio programa manje-više isti – ovisno o funkciji koja se poziva, program ispisuje drugačije rezultate.



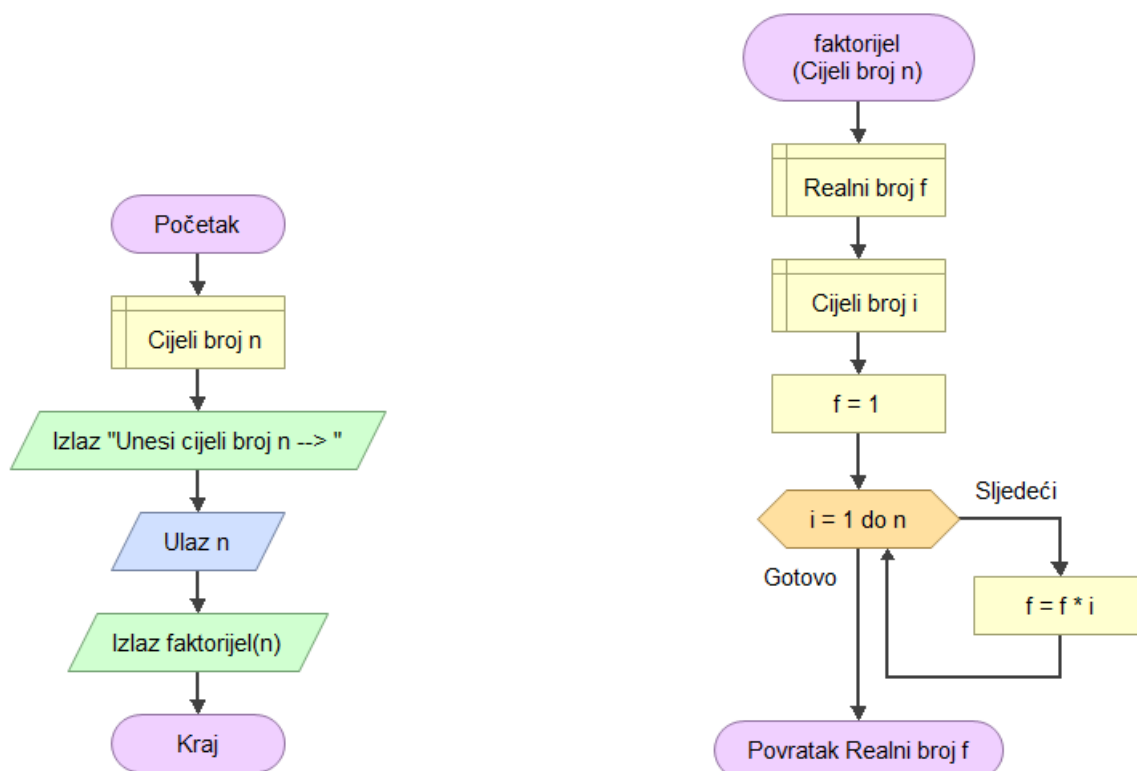
Očekivani izlaz:

2

3

6

Primjer 3 – Ovdje smisao funkcija dolazi do izražaja. U funkcije u pravilu izdvajamo složenije algoritme. Kombiniramo množenje i programsku petlju da bismo napisali program koji računa faktoriyel.



Očekivani izlaz:

Unesi cijeli broj →

5

120

Aktivnost 2 – u ovoj aktivnosti učenici samostalno rješavaju problem

Potrebno je samostalno izraditi program koji od korisnika traži unos broja. Taj broj nam govori koliko brojeva ćemo nakon njega unijeti. Za svaki uneseni broj potrebno je pozvati funkciju prima taj broj kao argument, a kao izlaznu vrijednost ispisuje je li broj paran ili neparan.

Očekivani izlaz:

Koliko brojeva unosiš? →

4

1

Neparan

2

Paran

3

Neparan

4

Paran

Metode za spoznajnu prilagodbu

1. Navesti učenika na sastavljanje popisa varijabli potrebnih za rješavanje zadatka

Primjer:

n – broj koji nam govori koliko brojeva će korisnik unijeti

broj – broj koji korisnik unosi u određenom koraku petlje

i – brojač koji nam govori u kojem koraku se petlja nalazi

2. Usmjeriti učenika na samostalno raspisivanje potrebnih algoritamskih konstrukata.

Primjer:

Grananje	Operacija utvrđivanja parnosti broja	Petlja
Potrebno nam je grananje da bismo utvrdili je li broj paran ili neparan	broj % 2 == 0 – broj je paran broj % 2 != 0 – broj je neparan	Uneseni broj n = 5 Korisnik unosi pet brojeva, potrebno je kreirati petlju koja se ponavlja pet puta

3. Definirati izlazne podatke programa

Primjer:

Za svaki uneseni broj program ispisuje poruku „Paran“ ili „Neparan“

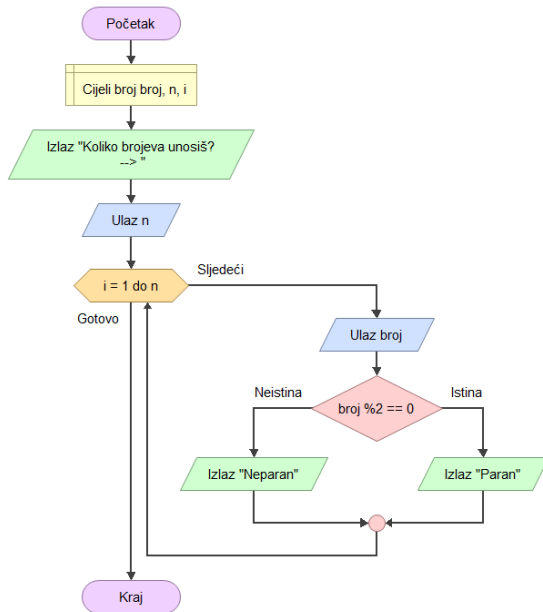
4. Raščlaniti zadatak na manje cjeline

Primjer:

Ulaz	Obrada	Izlaz
Unošenje broja n	Izvesti petlju n puta U svakom koraku petlje tražiti unos novog broja Testirati svaki broj na parnost	Za svaki uneseni broj unutar petlje ispisati poruku „Paran“ ili „Neparan“

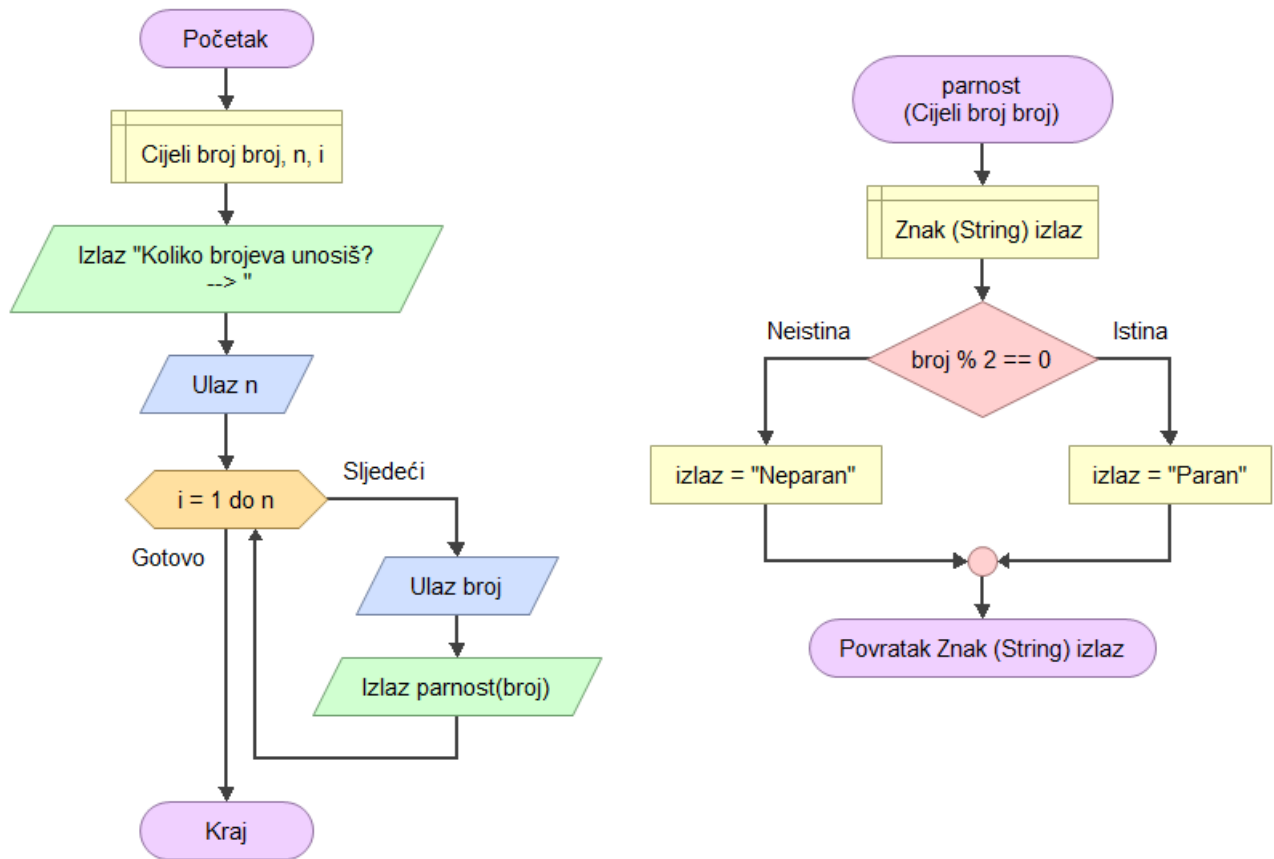
Metode za prilagodbu školskih zahtjeva:

1. Usmjerite učenika na samostalno rješavanje jednostavnijeg zadatka, na primjer pokušajte riješiti isti zadatak bez funkcije



2. Iskoristite rješenje jednostavnijeg zadatka kao model po kojem učenik s prilagodbom samostalno dolazi do rješenja glavnog zadatka. Ako je potrebno, uz pomoć nastavnika izdvojiti naredbe koje treba premjestiti u funkciju
3. Omogućite učeniku dulje vrijeme za rješavanje zadatka

Očekivano rješenje:



Vrednovanje

Učenik dobiva povratnu informaciju o usvojenosti novih sadržaja

Nastavnik dobiva povratnu informaciju o stupnju usvojenosti sadržaja od strane učenika

Poticanje natjecateljskog duha rješavanjem zadataka i uspoređivanje svojih odgovora s odgovorima drugih učenika

Vrednovanje za učenje:

Nastavnik prati aktivnosti tijekom rada

Učenik kroz razgovor s nastavnikom dobiva povratnu informaciju o svom načinu razmišljanja i tehnikama rješavanja problema

Nastavnik kroz odgovore učenika dobiva povratnu informaciju u kojoj mjeri je njegova aktivnost razumljiva te jesu li aktivnosti za učenike razumljivo postavljene

Vrednovanje kao učenje:

Učenici samovrednuju svoje odgovore

Učenici vrednuju odgovore drugih učenika

Evaluacija nastavnog sata

Poželjno je izraditi ili anketni listić ili online anketu u kojoj nastavnik dobiva anonimnu povratnu informaciju od učenika