Uvod u programske paradigme

Stefan Mišković

Matematički fakultet, Beograd

Programski jezici i programske paradigme

- **Programski jezik** je veštački jezik za pisanje instrukcija koje mogu biti prevedene u mašinski jezik i izvršene od strane računara.
- Postoji nekoliko hiljada programskih jezika.
- **Programska paradigma** (grč. obrazac, šablon) predstavlja stil programiranja.
- Postoji svega nekoliko vrsta programskih paradigmi.

Odnos paradigmi i programskih jezika

- Svakoj programskoj paradigmi pripada više programskih jezika.
- Jedan programski jezik može podržavati samo jednu paradigmu. Na primer, **C** podržava samo proceduralnu paradigmu.
- Jedan programski jezik može podržavati i više programskih paradigmi. Na primer, **C++** podržava proceduralni i objektno-orijentisani stil programiranja.

Vrste programskih paradigmi

- **Osnovne** programske paradigme:
  - proceduralna paradigma,
  - objektno-orijentisana paradigma,
  - funkcionalna paradigma,
  - logička paradigma.
- Ostale paradigme se često posmatraju kao potparadigme ili kombinacije osnovnih.

Proceduralna paradigma

- Procedurom se saopštava računaru **kako** se problem rešava.
- Navodi se precizan niz koraka za rešavanje problema.
- Primeri programskih jezika:
  - Pascal,
  - C,
  - Python...

Primer u C-u

```c
#include <stdio.h>

int main() {
    int broj = 24;
    printf("Faktori broja %d su: ", broj);
    for (int i = 1; i <= broj; i++) {
        if (broj % i == 0) {
            printf("%d ", i);
        }
    }
    return 0;
}
```

Primer u Pythonu

```
def aritmeticka_sredina(brojevi):
    suma = sum(brojevi)
    return suma / len(brojevi)

brojevi = [4, 8, 15, 16, 23, 42]
print("Aritmeticka sredina niza je:")
print(aritmeticka_sredina(brojevi))
```

Objektno-orijentisana paradigma

- Podaci i procedure se enkapsuliraju u **objekte**, koji su grupisani po klasama.
- **Klasa** je šablon na osnovu kog se kreiraju konkretni objekti.
- Objekti međusobno interreaguju razmenom poruka.
- Primeri programskih jezika:
  - Java,
  - C++,
  - Python...

Primer u Javi

```
public class Tacka {
    private int x, y;

    public Tacka(int x, int y) {
        this.x = x;
        this.y = y;
    }

    public void prikazi() {
        System.out.println("(" + x + ", " + y + ")");
    }

    public static void main(String[] args) {
        Tacka t = new Tacka(1, 2);
        t.prikazi();
    }
}
```

Primer u Pythonu

```
class Tacka:
    def __init__(self, x, y):
        self.x = x
        self.y = y

class Linija:
    def __init__(self, tacka1, tacka2):
        self.tacka1 = tacka1
        self.tacka2 = tacka2

    def duzina(self):
        dx = self.tacka2.x - self.tacka1.x
        dy = self.tacka2.y - self.tacka1.y
        return (dx ** 2 + dy ** 2) ** 0.5

t1 = Tacka(1, 2)
t2 = Tacka(4, 6)
linija = Linija(t1, t2)

print("Duzina linije:", linija.duzina())
```

Funkcionalna i logička paradigma

- Funkcionalna paradigma:
  - Izračunavanja se vrše evaluacijom **matematičkih funkcija**.
  - Najistaknutiji predstavnik: Haskell.
- Logička paradigma:
  - Zasnovana je na **matematičkoj logici**.
  - Koriste se aksiome, pravila izvođenja i upiti.
  - Najistaknutiji predstavnik: Prolog.

Primer u Haskellu

```
faktorijel :: Int -> Int
faktorijel 0 = 1
faktorijel n = n * faktorijel (n - 1)

zbirFaktorijela :: [Int] -> Int
zbirFaktorijela [] = 0
zbirFaktorijela (x:xs) = faktorijel x + zbirFaktorijela xs

main = print (zbirFaktorijela [1, 2, 3, 4, 5])
```

Primer u Prologu

- Proverava se da li je broj paran, tako što se oduzima broj 2, sve dok se ne dođe do 0.

```
paran(0).
paran(X) :- Y is X - 2, Y >= 0, paran(Y).
```

- Primer upita:

```
?- paran(4).
true.
```

Još neke paradigme

- Kod **skript paradigme** u programu se zadaje lista komandi koje mogu biti izvršene u zadatom okruženju.
  - Primeri primene su kod operativnih sistema i u izradi web aplikacija.
  - Primeri programskih jezika: JavaScript, TypeScript...
- **Paradigma upitnih jezika** se vezuje za programske jezike za baze podataka ili pronalaženje informacija.
  - Primer SQL upita:

```
SELECT Ime
FROM Studenti
WHERE Prezime='Petrovic';
```

- **Konkurentnu paradigmu** karakteriše više procesa koji se izvršavaju u istom vremenskom periodu, a imaju isti cilj.

Primer u JavaScriptu

```
function povrsinaKruga(r) {
    const pi = 3.14159;
    return pi * r * r;
}

const poluprecnik = 5;
const povrsina = povrsinaKruga(poluprecnik);

console.log("r = " + poluprecnik);
console.log("P = " + povrsina);
```

Jezici za obeležavanje

- Primeri: HTML, CSS, XML, JSON...
- Jezici za obeležavanje **nisu** programski jezici.
- Prema tome, oni ne pripadaju nijednoj programskoj paradigmi.