Matematički fakultet - OOP.

Objektno programiranje, vežbanje 05

Nemanja Mićović

1. Napraviti hijerarhiju klasa koja opisuje jednostavne artitmetičke izraze.

   - Izraz
     - Broj: `vrednost`
     - BinarnaOperacija: `levi operand, desni operand`
       * Sabiranje
       * Oduzimanje

   Klasa izraz treba da ima metod `double izracunaj()` koji vrši izračunavanje izraza.

   Klasa `BinarnaOperacija` u sebi od polja treba da sadrži referencu na dva izraza.

   Mala pomoć (ne znači da klase treba samo da sadrže navedeno):

```
class BinarnaOperacija extends Izraz {
    protected Izraz levi, desni;

    public BinarnaOperacija(Izraz levi, Izraz desni) {
        this.levi = levi;
        this.desni = desni;
    }
}

class Sabiranje extends BinarnaOperacija {
    public double izracunaj() {
        double vrednostLevogIzraza = levi.izracunaj();
        double vrednostDesnogIzraza = desni.izracunaj();
        return ...;
    }
}
```

   Na kraju, testirati sledeći kod u funkciji `main`:

```
public static void main(String[] args) {
    // 2 + 10 - 11 + 20 = 21
    Broj b2 = new Broj(2);
    Broj b10 = new Broj(10);
    Broj bn11 = new Broj(-11);
    Broj b20 = new Broj(20);

    Sabiranje s1 = new Sabiranje(b2, b10);
    Oduzimanje o1 = new Oduzimanje(s1, bn11);
    Sabiranje s2 = new Sabiranje(o1, b20);

    System.out.println("Rezultat: " + s2.izracunaj())
}
```

2. Proširiti hijerarhiju klasa za izraze tako da se omogući korišćenje promenljive.

   - Izraz
     - Broj: `vrednost`
     - BinarnaOperacija: `levi operand, desni operand`
       * Sabiranje
       * Oduzimanje
     - Promenljiva: `ime, vrednost`

   Implementirati i odgovarajući metod `String toString()` za svaki od listova u hijerarhiji.

   Na kraju, testirati sledeći kod u funkciji `main`

```
public static double[] vrednostPromenljive = {10, 20};

public static void main(String[] args) {
```

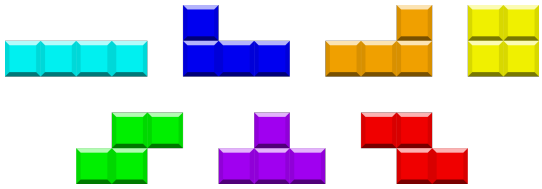
```
        // 2 + x0 - 11 + x1 = 21
        Broj b2 = new Broj(2);
        Promenljiva p0 = new Promenljiva("x1", 10);
        Broj bn11 = new Broj(-11);
        Promenljiva p1 = new Promenljiva("x2", 20);

        Sabiranje s1 = new Sabiranje(b2, p0);
        Oduzimanje o1 = new Oduzimanje(s1, bn11);
        Sabiranje s2 = new Sabiranje(o1, p1);

        System.out.println("Rezultat: " + s2.izracunaj())
    }
```

---

3. Napraviti hijerarhiju klasa koja opisuje blokove iz igre tetris. Blokovi koji postoje u igri Tetris su dati na sledećoj slici:



Blokove reprezentovati matricom karaktera. Omogućiti da metod `toString()` adekvatno prikazuje blokove. Implementirati funkcije za rotaciju bloka (rotirati matricu).

Za lepši ispis iz programa, korisiti neki od Unicode karaktera za blokove dostupne na adresi:

- `https://en.wikipedia.org/wiki/Block_Elements`

Dodatna motivacija za ovaj zadatak je da kod koji napišete, posle iskoristite za implementaciju prave igre Tetris!

4. Napraviti hijerarhiju klasa koje opisuje ljude na fakultetu. Osnovne klase koje treba implementirati (sa njihovim poljima):

- Čovek: `ime`, `prezime`, `godina rođenja`
  - Student: `trenutna godina studija`, `trenutni prosek`
    - Student osnovnih studija: `smer osnovnih studija`
    - Student master studija: `smer master studija`
  - Zaposleni: `godina zaposlenja`, `plata`
    - Nastavnik: `omiljeni predmet`, `listu predmeta na kojima drži nastavu`
      - Profesor: `titula (docent, vanredni, redovni)`
      - Asistent: `smer doktorskih studija`
    - Službenik: `odsek`

Obezbediti implementaciju funkcije `String toString()` koja adekvatno ispisuje objekte konstruisane hijerarhije.