

1. Napraviti hijerarhiju klasa koja opisuje jednostavne aritmetičke 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):

---

```
1 class BinarnaOperacija extends Izraz {
2     protected Izraz levi, desni;
3
4     public BinarnaOperacija(Izraz levi, Izraz desni) {
5         this.levi = levi;
6         this.desni = desni;
7     }
8 }
9
10 class Sabiranje extends BinarnaOperacija {
11     public double izracunaj() {
12         double vrednostLevogIzraza = levi.izracunaj();
13         double vrednostDesnogIzraza = desni.izracunaj();
14         return ...;
15     }
16 }
```

---

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

---

```
1 public static void main(String[] args) {
2     // 2 + 10 - 11 + 20 = 21
3     Broj b2 = new Broj(2);
4     Broj b10 = new Broj(10);
5     Broj bn11 = new Broj(-11);
6     Broj b20 = new Broj(20);
7
8     Sabiranje s1 = new Sabiranje(b2, b10);
9     Oduzimanje o1 = new Oduzimanje(s1, bn11);
10    Sabiranje s2 = new Sabiranje(o1, b20);
11
12    System.out.println("Rezultat: " + s2.izracunaj())
13 }
```

---

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`

---

```
1 public static double[] vrednostPromenljive = {10, 20};
2
3 public static void main(String[] args) {
```

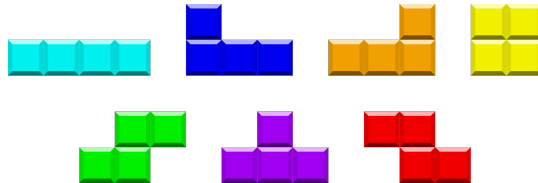
```

4      // 2 + x0 - 11 + x1 = 21
5      Broj b2 = new Broj(2);
6      Promenljiva p0 = new Promenljiva("x1", 10);
7      Broj bn11 = new Broj(-11);
8      Promenljiva p1 = new Promenljiva("x2", 20);
9
10     Sabiranje s1 = new Sabiranje(b2, p0);
11     Oduzimanje o1 = new Oduzimanje(s1, bn11);
12     Sabiranje s2 = new Sabiranje(o1, p1);
13
14     System.out.println("Rezultat: " + s2.izracunaj())
15 }

```

---

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, koristiti neki od Unicode karaktera za blokove dostupne na adresi:

- [https://en.wikipedia.org/wiki/Block\\_Elements](https://en.wikipedia.org/wiki/Block_Elements)

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