

# Programiranje I — 10. domača naloga

Rok za oddajo: nedelja, 21. januar 2018, ob 23:55

## Risar

### Naloga

Napišite razred `Risar` in hierarhijo razredov, izpeljanih iz razreda `Lik`, tako da bo s pomočjo metod objekta razreda `Risar` mogoče risati like (ti bodo predstavljeni kot objekti (podrazredov) razreda `Lik`) na koordinatno mrežo velikosti  $100 \times 100$ . Zgornja leva točka mreže ima koordinati  $(x, y) = (0, 0)$ , zgornja desna  $(99, 0)$ , spodnja desna pa  $(99, 99)$ .

Razrede definirajte kot javne statične notranje razrede (`public static class`) znotraj razreda `Oddaja`. Natančnejša navodila glede priprave datoteke za oddajo boste našli v razdelku `Oddaja`.

### Opis posameznih razredov

V nadaljevanju so navedeni konstruktorji in metode, ki jih boste implementirali v posameznih razredih. Attribute, pomožne metode, morebitne pomožne razrede in vsebino razreda `Lik` lahko določite povsem svobodno. Razrede `Pravokotnik`, `Mnogokotnik`, `Elipsa`, `Presek` in `Razlika` lahko izpeljete neposredno iz razreda `Lik`, lahko pa se domislite kakšne bolj razvejane hierarhije.

Pri vsakem razredu oziroma njegovem elementu so v oglatih oklepajih navedeni testni razredi, ki ga lahko uporabljajo.

### Razred `Pravokotnik` [J1–J11, S1–S50]

Razred `Pravokotnik` naj vsebuje sledeči konstruktor:

- `public Pravokotnik(int xLevo, int yZgoraj, int sirina, int visina)`

Ustvari objekt, ki predstavlja pravokotnik širine  $w = \text{sirina}$  in višine  $h = \text{visina}$  z zgornjim levim ogliščem na koordinatah  $x_0 = \text{xLevo}$  in  $y_0 = \text{yZgoraj}$ . Pravokotnik zaseda  $w$  točk mreže po širini in  $h$  po višini.

V vseh testnih primerih velja  $w \in [1, 100]$ ,  $h \in [1, 100]$ ,  $x_0 \in [0, 100 - w]$  in  $y_0 \in [0, 100 - h]$ .

### Razred `Elipsa` [J3–J11, S11–S50]

Razred `Elipsa` naj vsebuje sledeči konstruktor:

- `public Elipsa(int xSredisce, int ySredisce,  
int vodoravnaPolos, int navpicnaPolos)`

Ustvari objekt, ki predstavlja elipso s središčem v koordinatah  $x_s = \text{xSredisce}$  in  $y_s = \text{ySredisce}$  ter polosema  $a = \text{vodoravnaPolos}$  in  $b = \text{navpicnaPolos}$ . Elipsi pripadajo vse točke  $(x, y)$ , za katere velja

$$b^2(x - x_s)^2 + a^2(y - y_s)^2 \leq a^2b^2.$$

V vseh testnih primerih velja  $a \in [1, 49]$ ,  $b \in [1, 49]$ ,  $x_s \in [a, 99 - a]$  in  $y_s \in [b, 99 - b]$ .

### Razred Mnogokotnik [J5–J11, S21–S50]

Razred Mnogokotnik naj vsebuje sledeči konstruktor:

- `public Mnogokotnik(int[] [] omejitve)`

Ustvari objekt, ki predstavlja mnogokotnik. Mnogokotnik vsebuje vse točke  $(x, y)$ , ki izpolnjujejo sledeče omejitve:

$$\begin{aligned} a_1x + b_1y + c_1 &\leq 0 \\ a_2x + b_2y + c_2 &\leq 0 \\ &\dots \\ a_nx + b_ny + c_n &\leq 0 \end{aligned}$$

Prva vrstica tabele `omejitve` vsebuje parametre  $a_1$ ,  $b_1$  in  $c_1$  (v tem vrstnem redu), druga parametre  $a_2$ ,  $b_2$  in  $c_2$  itd.

V vseh testnih primerih velja  $n \in [3, 20]$  ter  $a_i \in [-100, 100]$ ,  $b_i \in [-100, 100]$  in  $c_i \in [-10^4, 10^4]$  za vse  $i \in \{1, \dots, n\}$ . Poleg tega so omejitve določene tako, da se mnogokotnik v celoti nahaja znotraj koordinatne mreže  $100 \times 100$ .

### Razred Presek [J6–J11, S26–S50]

Razred Presek naj vsebuje sledeči konstruktor:

- `public Presek(Lik prvi, Lik drugi)`

Ustvari objekt, ki predstavlja presek likov `prvi` in `drugi`. Presek vsebuje vse točke, ki pripadajo obema likoma hkrati.

### Razred Razlika [J7–J11, S31–S50]

Razred Razlika naj vsebuje sledeči konstruktor:

- `public Razlika(Lik prvi, Lik drugi)`

Ustvari objekt, ki predstavlja razliko likov `prvi` in `drugi`. Razlika vsebuje vse točke, ki pripadajo liku `prvi`, obenem pa ne pripadajo liku `drugi`.

### Razred Risar [J1–J11, S1–S50]

Razred Risar naj vsebuje sledeče elemente:

- `public Risar()` [J1–J11, S1–S50]

Ustvari objekt razreda `Risar`.

- `public void narisiLik(Lik lik)` [J1–J11, S1–S50]

Na koordinatno mrežo doda sliko lika `lik`.

- `public void narisiRob(Lik lik, int debelina)` [J8–J11, S36–S50]

Na koordinatno mrežo doda sliko roba lika `lik`. Parameter `debeline` podaja debelino roba. Točka  $(x, y)$  pripada robu debeline 1 natanko tedaj, ko pripada liku, obenem pa vsaj ena od njenih štirih sosed (leva, zgornja, desna, spodnja) ne pripada liku. Točka  $(x, y)$  pripada robu debeline  $d > 1$  natanko tedaj, ko pripada liku, poleg tega pa je izpolnjen še vsaj eden od sledečih pogojev:

- točka  $(x, y)$  pripada robu debeline  $d - 1$ ;
- vsaj ena od štirih sosed točke  $(x, y)$  pripada robu debeline  $d - 1$ .

Parameter `debeline` je v testnih primerih J8 in S36–S40 enak 1, v primerih J9–J11 in S41–S50 pa se nahaja znotraj intervala  $[1, 50]$ .

Na sliki 1 sta prikazana rob elipse z debelino 3 in rob mnogokotnika z debelino 4. Rdeče točke tvorijo rob debeline 1. Rdeče in zelene točke skupaj tvorijo rob debeline 2. Rob debeline 3 je sestavljen iz rdečih, zelenih in modrih točk, rob debeline 4 pa iz rdečih, zelenih, modrih in vijoličastih točk.

- `public boolean[][] slika()` [J1–J11, S1–S50]

Vrne tabelo velikosti  $100 \times 100$ , ki predstavlja sliko na koordinatni mreži. Element v vrstici z indeksom  $i$  in stolpcu z indeksom  $j$  naj ima vrednost `true` natanko v primeru, če točka s koordinatama  $x = j$  in  $y = i$  pripada sliki (torej vsaj enemu od narisanih likov oziroma robov).

## Testni primer J9

Testni razred:

```
public class Test09 {

    public static void main(String[] args) {
        Oddaja.Risar risar = new Oddaja.Risar();

        Oddaja.Lik elipsa1 = new Oddaja.Elipsa(7, 5, 6, 4);
        Oddaja.Lik elipsa2 = new Oddaja.Elipsa(22, 5, 6, 4);

        // trikotnik z oglišči (2,29), (26,29), (14,8)
        Oddaja.Lik trikotnik = new Oddaja.Mnogokotnik(
            new int[][]{
                { 0, 1, -29 },
                { 7, -4, -66 },
                { -7, -4, 130 }
            }
        );
    }
}
```

```

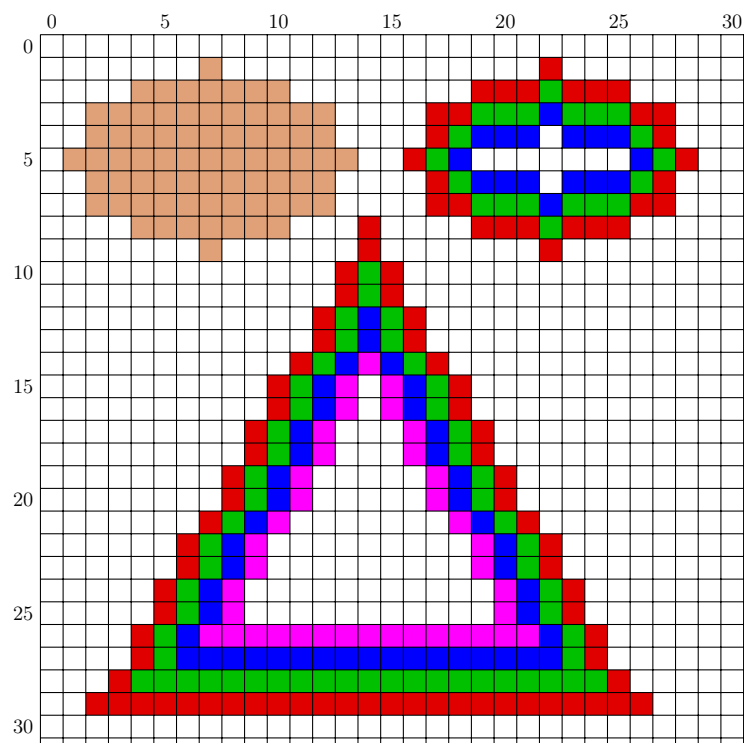
    risar.narisiLik(elipsa1);
    risar.narisiRob(elipsa2, 3);
    risar.narisiRob(trikotnik, 4);

    prikazi(risar.slika());
}

private static void prikazi(boolean[][] slika) {
    for (int i = 0; i < slika.length; i++) {
        for (int j = 0; j < slika[i].length; j++) {
            System.out.print(slika[i][j] ? "*" : "- ");
        }
        System.out.println();
    }
}
}

```

Gornji testni razred izpiše vsebino celotne koordinatne mreže. Točke, ki pripadajo sliki, so predstavljene z znaki \*, ostale točke pa z znaki -. Zgornji levi del koordinatne mreže je grafično prikazan na sliki 1. Različne barve smo uporabili le za lažje razumevanje robov.



Slika 1: Grafični prikaz zgornjega levega odseka slike, ki jo ustvari testni razred J9.

## Oddaja naloge

Oddajte datoteko Oddaja.java. Datoteka naj ima sledečo strukturo:

```

// 63170999 (Zamenjajte s svojo vpisno številko!)

public class Oddaja {

```

```
public static class Lik {  
    ...  
}  
  
public static class Pravokotnik {  
    ...  
}  
  
...  
  
public static class Risar {  
    ...  
}  
}
```

V prikazani strukturi manjkajo določila, ki se nanašajo na dedovanje (razred `Lik`, denimo, bi lahko bil abstrakten, razred `Pravokotnik` bo verjetno podrazred nekega drugega razreda itd.). Notranjih razredi si lahko sledijo v poljubnem vrstnem redu. Seveda lahko dodate tudi kak svoj razred, lahko pa, če se boste zadovoljili z manjšim številom točk, katerega od predpisanih razredov tudi izpustite.