

# Vaje pri predmetu Programiranje 1

## Teden 11: Generiki in vmesniki

### Liki in primerjalniki

#### Naloga

Pri tej nalogi bomo izhajali iz malenkostno spremenjenih razredov, ki tvorijo hierarhijo likov (atributi ter rezultati metod `ploscina` in `obseg` so zaradi enostavnosti tipa `int` namesto `double`), iz razreda `Vektor` in iz razreda `Skupno`, ki vsebuje sledeči metodi:

```
public static <T extends Comparable<T>> void uredi(Vektor<T> vektor)
public static <T> void uredi(Vektor<T> vektor, Comparator<T> primerjalnik)
```

Vse navedeno smo spoznali na predavanjih.

Rešite sledeče naloge:

- Razred `Lik` dopolnite tako, da bo implementiral vmesnik `Comparable<Lik>`. Za potrebe te naloge naj bo naravna urejenost likov določena takole: lik *A* sodi pred lik *B* natanko tedaj, ko ima lik *A* manjšo ploščino od lika *B*.

- V razred `Lik` dodajte metodo

```
public static Comparator<Lik> poObsegu(),
```

ki vrne primerjalnik, ki primerja podana lika po obsegu. Lik z manjšim obsegom naj se nahaja pred likom z večjim obsegom.

- V razred `Lik` dodajte metodo

```
public static Comparator<Lik> poTipu(),
```

ki vrne primerjalnik, ki primerja podana lika po tipu. Pravokotniki naj bodo pred kvadrati, ti pa pred krogi. (Nalogo lahko elegantno rešite tako, da v hierarhijo likov dodate metodo, ki vrne celoštevilsko konstanto, ki je odvisna od tipa objekta `this`.)

- V razred `Skupno` dodajte metodo

```
public static <T> Comparator<T> kompozitum(Comparator<T> prim1,
                                           Comparator<T> prim2),
```

ki vrne primerjalnik, pri katerem metoda `compare` vrne rezultat medsebojne primerjave podanih objektov s primerjalnikom `prim1`, če se objekta po tem kriteriju razlikujeta, če se ne, pa vrne rezultat primerjave objektov s primerjalnikom `prim2`.

- V razred `Lik` dodajte metodo

```
public static void urediPoTipuInObsegu(Vektor<Lik> vektor),
```

ki like v podanem vektorju uredi po tipu (najprej pravokotniki, nato kvadrati, nazadnje krogi), like istega tipa pa po naraščajočem obsegu. Pomagajte si z metodo `kompozitum`.