Teemad. *Pärilus Javas. Ülem- ja alamklassid. Ülekatmine.*

Pärast selle praktikumi läbimist oskab üliõpilane

- luua alamklasse;
- kasutada ülemklassi meetodeid;
- katta üle ülemklassi meetodeid;
- kasutada võtmesõna super.

Praktikumijuhend

Tahtes olemasolevale klassile lisada omadusi (meetodid, väljad), kuid seda klassi ennast mitte muuta, võime luua uue klassi, milles kirjeldame lisatavad meetodid ja väljad. Niiviisi loodud klassi nimetame alamklassiks ja klassi, millest lähtusime, ülemklassiks. Sellist tegevust võib üldjuhul korduvalt jätkata.

Ülemklassi omadused kanduvad üle tema alamklassidele. Vastavat mehhanismi nimetatakse päriluseks. Pärilus on üks objektorienteeritud programmeerimise alustalasid, võimaldades loodud klasside taaskasutamist ja tekitades klasside hierarhilise struktuuri.

Märkimaks, et klass B on klassi A alamklass, tuleb klassi nime järele lisada võtmesõna extends ja ülemklassi nimi:

```
class A {
  . . .
}

class B extends A {
  . . .
}
```

Alamklassis saab kasutada kõiki neid ülemklassi muutujaid ja meetodeid, mille piiritlejaks ei ole private. Sama signatuuriga meetodi uuestikirjeldamist alamklassis nimetatakse meetodi ülekatmiseks.

Olgu meil klass Kujund, mis esitab tasandil asuvate geomeetriliste kujundite üldise kirjelduse. Klassis Kujund on kaks isendivälja: x ja y. Nende väärtused määravad üheselt kujundi asukoha tasandil.

```
class Kujund {

  int x;
  int y;

  Kujund() {
    x = 0;
    y = 0;
  }

  Kujund(int xKoordinaat, int yKoordinaat) {
    x = xKoordinaat;
    y = yKoordinaat;
  }
}
```

Samuti olgu meil klass Ruut, mis kirjeldab ruutu tasandil ja on klassi Kujund alamklass. Klassi Ruut puhul tähistagu isendiväljad x ja y ruudu vasaku ülemise nurga koordinaate, millega on üheselt määratud ruudu paiknemine tasandil.

```
class Ruut extends Kujund {

  int külg;

  Ruut(int xKoordinaat, int yKoordinaat, int küljepikkus) {
    x = xKoordinaat;
    y = yKoordinaat;
    külg = küljepikkus;
  }
}
```

Vaatleme veel klassi Ring, mis kirjeldab ringi tasandil ja on samuti alamklass klassile Kujund. Alamklassi Ring puhul tähistagu isendiväljad x ja y ringi keskpunkt koordinaate.

```
class Ring extends Kujund {

  int raadius;

  Ring(int xKoordinaat, int yKoordinaat, int r) {
    x = xKoordinaat;
    y = yKoordinaat;
    raadius = r;
  }
}
```

Nimetame ankurpunktiks kujundit täpselt ümbritseva ristküliku ülemist vasakut nurka. Ruudu puhul on selleks ruudu enda ülemine vasak nurk. Ringi puhul võrdub ankurpunkti x-koordinaat keskpunkti x-koordinaadi ja raadiuse vahe ning y-koordinaat keskpunkti y-koordinaadi ja raadiuse vahega. Klassis Kujund eeldame, et ankurpunkt langeb kokku sama punktiga, mida tähistavad olemasolevad väljad x ja y.

Lisame klassile Kujund veel meetodi kujundi ümbermõõdu leidmiseks. Klass Kujund ise on üldistus, seega ei saa see meetod tagastada muud kui arvu 0. Lisame ka meetodi toString(), mis tagastab klassi Kujund isendi kirjelduse sõnena.