

Utrjevanje:

- Osnove OOP:
- Dedovanje, ponovna uporabljivost

Naloga 1.

Deklarirajte razred `AstralnoTelo`, ki vsebuje lastnost 'naziv' in konstruktor.

Iz njega izpeljite razred `Zvezda`, ki deduje lastnost `naziv` (pazite na opredelitev kvalifikatorja za dostopne pravice v razredu `AstralnoTelo`) tako, da je del razreda `Zvezda`. V razredu `Zvezda` definirajte še zaščitene (protected) lastnosti: `nazivZvezde`, `radij` in `oddaljenostOdSonca`.

`Zvezda` naj vsebuje tudi konstruktor (lahko tudi več konstruktorjev), ter primerjalne metode:

- razredna metoda `vrniSoncuBlizjoZvezdo/2` dobi kot argument 2 zvezdi in vrne tisto, ki je bližje soncu;
- razredna metoda `vrniManjsoRazdaljo/2` vrne razdaljo tiste z argumentum podane zvezde, katere razdalja do Sonca je manjša;
- objektno metodo `manjsaOdRazdalj/1` primerja oddaljenost zvezde in z argumentum podane in vrne manjšo od obeh.

Razred naj vsebuje še metodo, ki zna sešteti in tudi sešteje dve Zvezdi - Zvezdi seštejemo tako, da radiju večje zvezde dodamo 10% radija manjše; oddaljenost postane oddaljenost bližje zvezde; naziv nove zvezde pa se sestavi iz prvega znaka večje in prvega znaka manjše zvezde. **Če sta radija obeh zvezd enaka, zvezdi ne moremo in ju NE seštejemo!**

V zagonski metodi demonstracijskega razreda (lahko je to tudi razred `Zvezda`) ustvarite 5 zvezd in jih inicializirajte (vpiši podatke). Programsko poiščite zvezdo **najbližjo** Soncu in jo izpišite. Nato prvi definirani zvezdi prištejte vse preostale zvezde, ki jih lahko (različna radija) in izpišite končni radij novonastale zvezde.

Izpis zvezde naj se izvede kot izpis n-členika z nazivom zvezde na začetku, ki mu sledijo vsi preostali atributi oz. lastnosti zvezde.

Naloga 2.

Zapišite definicijo razreda `Racunalnik`, ki vsebuje naslednje lastnosti: vrsto CPE, velikost trdega diska, velikost pomnilnika ter cena. Dodajte konstruktor z inicializacijo lasnosti in metode za vnos in vračanje njegovih lastnosti (setterje in getterje).

Iz razreda `Racunalnik` izpeljite razred `Prenosnik`, z dodatnimi lastnostmi velikost zaslona ter trajanje (čas avtonomije) baterije. Razred `Prenosnik` naj vsebuje konstruktor, setterje in getterje ter metodo, ki vrne velikosti zaslona v cm (vrednost lastnosti velikost zaslona je zapisana v palcih).

Zagonska metoda razreda `Računalnik` naj služi kot ovojnica za preskus definicij in delovanja: V njej definirajte, skreirajte in inicializirajte dva prenosnika (objekta razreda `Prenosnik`), ter izvedite demonstracijo rabo vseh metod. Na koncu izpiši, kateri izmed obeh prenosnikov ima boljše razmerje med trajanjem baterije in ceno.

Po potrebi v definicijo razredov lahko dodate še kakšno metodo.

Naloga 3.

Ustvariti bi želeli YesNo gumb za JavaFX aplikacijo. Objekt vrste YesNo pri postavitvi na sceno prikazuje napis Yes, klik na gumbu zamenja vsebino napisa na No, vsak naslednji ponovno zamenja Yes z No oz. No z Yes.

Obstoječi gumb vrste `javafx.scene.control.Button` je funkcionalno poln gumb, le da ima zgolj eno fiksno lastnost `'text'` za prikazovanje vrednosti na gumbu.

Pripravimo test dostopa do lastnosti `'text'` gumba ob kliku:

Uporabimo obstoječo JavaFX aplikacijo, ki jo lahko naredimo z čarovnikom v BlueJ. Poiščemo servisno metodo, ki se izvede ob kliku na gumb in kot zadnjo vrstico dodamo:

```
Button bb = (Button)event.getSource(); bb.setText("trilili");
```

- a) spišite mehanizem preklapljanja med Yes in No. Gumb naj bo vrste Button.
- b) Na osnovi Button kreirajte razred YesNo Button. Konstruktor za novi gumb prejme dve besedi (Yes, No) in pri klikanju naj preklaplja med njima (preklaplja med poljubnima podanima dvema besedama). Dodajte še oba 'fall back' konstruktorja; če gumb naredite zgolj z eno besedo, je druga vedno 'N-e', če ga ustvarite brez, pa je prva 'Da', druga 'Ne'.