Programiranje 1 — deveta domača naloga

Rok za oddajo: nedelja, 29. decembra 2024

Divjina

Uvod

Ta naloga je nekoliko nenavadna. Morda je tudi nekoliko težja. Kakorkoli že, če se je boste kljub temu lotili, se nikar ne zadovoljite s programom, v katerem mrgoli operatorjev `instanceof` in podobnih grdobij, pa tudi če pravilno obravnava vse testne primere. Ne, ne in ne! Zasledujte lepoto: vaš cilj naj bo rešitev, ki je karseda elegantna in po možnosti tudi kratka. Kaj vam bo 100%, če pa se vam bo ob slehernem pogledu na lasten program obrnil želodec?

Ozadje

V divjini je življenje enostavno. Živali se bodisi prehranjujejo bodisi preganjajo druga drugo. Recimo: želva se prehranjuje. Ali pa: tiger preganja medveda. Da pa vse skupaj le ne bo preveč trivialno, vas moramo opozoriti, da se v času, ko se prehranjuje želva, prehranjuje tudi plazilec — preprosto zato, ker je želva poseben primer plazilca. Seveda se sočasno prehranjuje tudi žival, saj plazilec ni nič drugega kot poseben primer živali. Ko zlobni tiger preganja ubogega medveda, je slika še nekoliko zanimivejša, saj obenem

- mačka preganja medveda (tigri so mačke, kot vam znajo povedati že v otroškem vrtcu);
- sesalec preganja medveda (mačke so sesalci);
- žival preganja medveda (sesalci so živali);

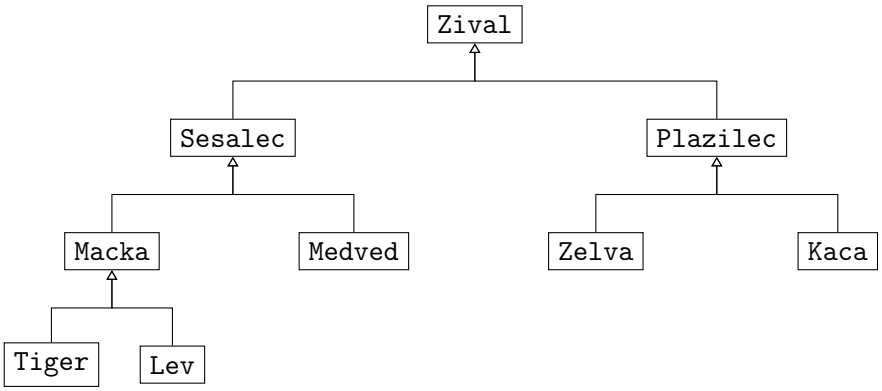
... in seveda ...

- tiger preganja sesalca (tudi medvedje so sesalci);
- mačka preganja sesalca;
- sesalec preganja sesalca;
- žival preganja sesalca;

... in seveda ...

- tiger preganja žival;
- mačka preganja žival;
- sesalec preganja žival;
- žival preganja žival.

Sedaj že slutimo, da bo živalski svet divjine predstavljen z razredi, ti pa bodo organizirani v hierarhijo. Kot prikazuje slika 1, je naša hierarhija sestavljena iz reci in piši devet razredov. Nobeden od njih ne bo abstrakten, poleg tega pa bodo njihovi objekti imeli drugačen pomen, kot smo ga bili vajeni doslej. Po običajni interpretaciji objektov bi, denimo, objekta `t` in `u`, ki ju ustvarita stavka



Slika 1: *Hierarhija razredov za predstavitev favne v divjini.*

```
Tiger t = new Tiger();
Tiger u = new Tiger();
```

predstavljala dva različna konkretna tigra, recimo samca po imenu Rjovko in samico po imenu Šapka. Tokrat ne bo tako. Vsak objekt tipa `Tiger` bo predstavljal *poljubnega* tigra. Na primer, klic metode

tiger`.seHrani()`

pri čemer je *tiger* kazalec na objekt tipa `Tiger`, predstavlja dejstvo, da se *nek* tiger hrani, a nas ne zanima, kateri. S stavki

```
t1.seHrani()
t2.seHrani()
t3.seHrani()
```

pri čemer kazalci `t1`, `t2` in `t3` kažejo na objekte tipa `Tiger`, potemtakem povemo samo to, da smo trikrat opazili *nekega* tigra, kako se hrani, pri tem pa nam je vseeno, ali gre za tri različne tigre ali pa za enega in istega, in to *ne glede na identitete objektov*. Ni važno, ali so objekti, na katere kažejo kazalci `t1`, `t2` in `t3`, isti ali ločeni; zanima nas samo to, da gre za objekte tipa `Tiger`. Vsak klic metode `seHrani` nad katerimkoli objektom tipa `Tiger` podaja dejstvo, da si nek (katerikoli) tiger (in s tem tudi mačka in sesalec in žival) teši lakoto.

Naloga

Napišite razrede `Zival`, `Sesalec`, `Macka`, `Tiger`, `Lev`, `Medved`, `Plazilec`, `Zelva` in `Kaca`, po potrebi pa lahko dodate še kak svoj razred. Hierarhijo definirajte tako, da bo na vseh njenih objektih mogoče klicati sledeče metode:

- `public void seHrani()`

  Ta metoda se pokliče, ko opazovalec opazi, da se prehranjuje žival, katere tip je določen s tipom objekta `this` (»objekt `this`« je, kot vemo, samo okrajšava za »objekt, na katerega kaže kazalec `this`«). Če se, denimo, prehranjuje želva, se bo metoda poklicala kot

  zelva`.seHrani();`

  pri čemer je *zelva* kazalec na objekt tipa `Zelva`, če se prehranjuje žival, o kateri vemo samo to, da je sesalec, pa bomo metodo poklicali kot

  sesalec`.seHrani();`

pri čemer je *sesalec* kazalec na objekt tipa `Sesalec`.

- `public void preganja(Zival druga)`

  Ta metoda se pokliče, ko opazovalec opazi, da žival, katere tip je določen s tipom objekta `this`, preganja žival, katere tip je določen s tipom objekta `druga`. Če, denimo, mačka preganja plazilca, bomo metodo poklicali kot

  macka`.preganja(`*plazilec*`);`

  pri čemer je *macka* kazalec na objekt tipa `Macka`, *plazilec* pa kazalec na objekt tipa `Plazilec`.

- `public int steviloHranjenj()`

  Vrne podatek o tem, kolikokrat je opazovalec opazil, da se prehranjuje žival, katere tip je določen s tipom objekta `this`.

- `public int steviloPreganjanj(Zival druga)`

  Vrne podatek o tem, kolikokrat je opazovalec opazil, da žival, katere tip je določen s tipom objekta `this`, preganja žival, katere tip je določen s tipom objekta `druga`.

Testni primeri

V skritih testnih primerih 1–25 se kličeta samo metodi `seHrani` in `steviloHranjenj`. V primerih 1–10 in 26–35 so objekti, nad katerimi se kličejo metode `seHrani`, `preganja`, `steviloHranjenj`, `steviloPreganjanj`, in objekti, ki jih metodi `preganja` in `steviloPreganjanj` prejmeta kot parameter, lahko le tipa `Tiger`, `Lev`, `Medved`, `Zelva` ali `Kaca`.

Javni testni primer

Sledeča koda je del edinega javnega testnega razreda:

```
Zival zival = new Zival();
Sesalec sesalec = new Sesalec();
Macka macka = new Macka();
Tiger tiger = new Tiger();
Lev lev = new Lev();
Medved medved = new Medved();
Plazilec plazilec = new Plazilec();
Zelva zelva = new Zelva();
Kaca kaca = new Kaca();

tiger.seHrani();
medved.seHrani();
zelva.seHrani();
plazilec.seHrani();

System.out.println(zelva.steviloHranjenj());     // 1
System.out.println(kaca.steviloHranjenj());      // 0
System.out.println(plazilec.steviloHranjenj()); // 2
System.out.println(macka.steviloHranjenj());     // 1
System.out.println(sesalec.steviloHranjenj());   // 2
System.out.println(zival.steviloHranjenj());     // 4
System.out.println("---------");
```

```
tiger.preganja(medved);
sesalec.preganja(kaca);
zelva.preganja(zival);
tiger.preganja(medved);
lev.preganja(lev);
plazilec.preganja(lev);

System.out.println(tiger.steviloPreganjanj(medved));      // 2
System.out.println(macka.steviloPreganjanj(lev));         // 1
System.out.println(macka.steviloPreganjanj(sesalec));     // 3
System.out.println(macka.steviloPreganjanj(plazilec));    // 0
System.out.println(zelva.steviloPreganjanj(sesalec));     // 0
System.out.println(sesalec.steviloPreganjanj(plazilec)); // 1
System.out.println(plazilec.steviloPreganjanj(zival));    // 2
System.out.println(zival.steviloPreganjanj(plazilec));    // 1
System.out.println(zival.steviloPreganjanj(zival));       // 6
```

Oddaja naloge

Oddajte najmanj devet datotek, po eno za vsak razred iz hierarhije. Po potrebi lahko oddate še več razredov, a v skupnem seštevku ne več kot 20. V nasprotju z običajno prakso tokrat v imena razredov nikar ne tlačite svoje vpisne številke!