

Programiranje 2 - pisni izpit

7. 6. 2024 ob 10.00 (čas pisanja: 100 minut)

Naloga 1 (25 točk)

Definirajmo *palindrom* kot niz, sestavljen iz malih črk angleške abecede brez presledkov, ki se enako bere v obe smeri (naprej in nazaj). Primer: cepec, abba, lepobopel, navijajivan.

V razredu `Naloga1` napiši dve metodi: `boolean jePalindrom(String niz)` in `boolean jePalindromR(String niz)`, ki vrneta `true`, če je podani niz palindrom in `false` sicer. Prva metoda naj bo statična in naj problem reši na iterativni način (z zanko), druga pa naj bo ne-statična in rekurzivna.

Napiši tudi metodo `main()`, v njej pokliči obe metodi (kot parameter podaj prvi argument ob klicu programa) in s pomočjo tridelnega operatorja `?:` izpiši rezultat vsake od njiju kot »da« ali »ne«.

Primer: ob klicu `java Naloga1 cepec` naj program na izhod izpiše

da
da

ob klicu `java Naloga1 test` pa naj program izpiše

ne
ne

Pred oddajo preveri pravilnost delovanja na [spletnem preverjevalniku](#).

Naloga 2 (25 točk)

Napiši razred `Naloga12` in v njem metodo `void preberiInIzpisi(String imeDatoteke)`, ki prebere vsebino tekstovne datoteke `imeDatoteke` in izpiše 5 črk angleške abecede (A-Z), ki so v tej datoteki najpogostejše uporabljene, ter število njihovih pojavitev. Pri tem ne razlikuj med malimi in velikimi črkami. Izpis naj bo urejen po naraščajočem številu pojavitev.

Napiši tudi metodo `main()` in v njej kliči metodo `preberiInIzpisi()` s prvim argumentom, ki je podan ob klicu programa `Naloga12`.

Primer: ob klicu `java Naloga12 besediloS.txt` naj program izpiše

```
1. E : 63
2. A : 55
3. S : 42
4. O : 40
5. N : 39
```

Opomba: za 15 točk je dovolj, da izpišeš vse črke (od A do Z) in njihove pojavitve v poljubnem vrstnem redu.

Pred oddajo preveri pravilnost delovanja na [spletnem preverjevalniku](#).

Naloga 3 (25 točk)

Pri igrici Ladjice potapljat dva igralca vsak na svoje polje velikosti $v \times s$ (višina x širina) postavita po N ladij, nato pa izmenično ugibata koordinate nasprotnikovih ladij. Zmaga tisti, ki ugame koordinate vseh nasprotnikovih ladij. Pri postavitvi ladij morata igralca upoštevati osnovno pravilo, in sicer da se postavljene ladje med seboj ne smejo dotikati (niti v eni točki).

V datoteki `Naloga13.java` napiši razred `Ladjice` za pomoč pri igranju opisane igre. Razred naj vsebuje:

- atribut `int [][] igralnoPolje`, v katerem hrani postavitev ladjic (vrednost 0 – ladjice ni, vrednost 1 – ladjica je);
- konstruktor `Ladjice(int v, int s, int[] ladje)`, pri čemer sta v in s višina in širina igralnega polja, `ladje` pa seznam števila ladij posamezne velikosti (na primer, vrednost `ladje={4,2,5,1}` pomeni, da igralec na igralno polje postavi 4 ladje velikosti 1, 2 ladij velikosti 2, 5 ladij velikosti 3 in 1 ladjo velikosti 4);
- metodo `postaviLadjice()`, ki postavi ladjice naključno na igralno polje (pri tem upošteva število ladjic posamezne velikosti, kot je zapisano v atributu `ladje`, podanem v konstruktorju); za pisanje te metode glej dodatno navodilo spodaj;
- metodo `izpisiIgralnoPlosco()` za izpis postavljenih ladjic.

V razredu `Ladjice` naj bodo vsi atributi privatni, dostop do njih omogoči preko getter/setter metod.

Napiši tudi razred `Naloga13` in v njem metodo `main()` takole:

```
public static void main(String[] args) {  
    Ladjice ip = new Ladjice(10, 10, new int[]{4,3,2,1});  
    ip.postaviLadjice();  
    ip.izpisiIgralnoPlosco();  
}
```

Primer: Če bo metoda `postaviLadjice()` ladjice postavila takole

```
(vrstica, stolpec, dolzina, smer)  
(0,0,2,vodoravno), (0,6,2,navpično),  
(1,3,4,navpično), (2,1,1,navpično),  
(4,7,1,navpično), (4,9,1,navpično),  
(6,0,3,navpično), (6,2,2,navpično)  
(7,8,3,navpično), (8,5,1,navpično),
```

potem naj metoda `izpisiIgralnoPlosco()` izpiše, kot je prikazano na desni sliki.

Za izpis okvira uporabi znake : `┌` (`\u250c`), `┐` (`\u2510`), `└` (`\u2514`),
`┘` (`\u2518`), `─` (`\u2500`) in `|` (`\u2502`)

* *		*
	*	*
* *		
	*	
	*	*
		*
* *		
* *		*
*		*
		*

Dodatno navodilo: za izvedbo metode `postaviLadjice()` si napiši naslednje pomožne metode:

- `lahkoDodam(int i, int j, int smer, int dolzina)` ...preveri, ali lahko na koordinato (i, j) dodam ladjico velikosti `dolzina` v podani smeri (0 ... navpično, 1 ... vodoravno); pri tem si lahko pomagaš z metodo `devetProstih(int i, int j)`, ki preveri, če so polje (i, j) in vsa njegova sosedna polja prosta (metodo najdeš na eUčilnici);
- `dodajLadjico(int dolzina)` ... naključno ustvari koordinate i in j ter smer, nato pa preveri, ali sme dodati tako ladjo; če sme, jo doda, sicer ponovi (ponovno ustvari koordinate, ...);
- `postaviLadjice()` ... za vse ladjice, kot jih določa tabela `ladje`, kliče metodo `dodajLadjico()`.

Naloga 4 (25 točk)

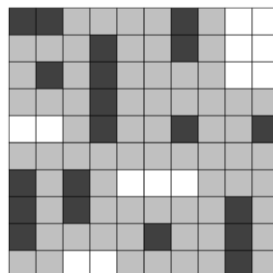
V razredu `Naloga14` napiši metodo `narisiLadjice(int[][] igralnoPolje)`, ki s pomočjo knjižnice `StdDraw` (datoteka `stdlib.jar` je na eUčilnici) nariše igralno ploščo in na njej ladjice, kot določa tabela `igralnoPolje`. Na poljih, kjer je na igralnem polju postavljena ladjica (vrednost 1) nariši temno siv kvadrateg; na poljih, ki so v bližnji okolici ladjic (vsa polja, ki se dotikajo ladjic), nariši svetlo siv kvadrateg, prazna polja pusti nepobarvana (belo).

Napiši tudi metodo `main()` in v njej klic metode `narisiLadjice()`, na primer takole:

```
narisiLadjice(new int[][]{
    {1, 1, 0, 0, 0, 0, 1, 0, 0, 0},
    {0, 0, 0, 1, 0, 0, 1, 0, 0, 0},
    {0, 1, 0, 1, 0, 0, 0, 0, 0, 0},
    {0, 0, 0, 1, 0, 0, 0, 0, 0, 0},
    {0, 0, 0, 1, 0, 0, 1, 0, 0, 1},
    {0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
    {1, 0, 1, 0, 0, 0, 0, 0, 0, 0},
    {1, 0, 1, 0, 0, 0, 0, 0, 1, 0},
    {1, 0, 0, 0, 0, 1, 0, 0, 1, 0},
    {0, 0, 0, 0, 0, 0, 0, 0, 1, 0}
});
```

(če si uspešno rešil nalogo 3, lahko uporabiš tudi tabelo, ki jo ustvari metoda `postaviLadjice()`).

Zgornja metoda naj izriše tako igralno polje:



Nasvet: za pomoč pri barvanju sosednjih polj napišite pomožno metodo `boolean jeOkoljca(int i, int j, int [][] polje)`, ki preveri, ali je v katerem od sosednjih polj polja (i, j) postavljena ladjica.