Tekmovanje programov za igro Minolovec

1 Osnovni podatki

Vabimo vas k udeležbi na tekmovanju programov za igro Minolovec. Če želite sodelovati, oddajte svoj program na Učilnico do **nedelje, 10. januarja 2021**. Avtorji najboljših programov bodo na izpitu nagrajeni z dodatnimi točkami.

2 Minolovec

Igra Minolovec (angl. Minesweeper)[1] se odvija na pravokotni plošči, sestavljeni iz $h \times w$ (h je višina, w pa širina) kvadratnih polj. Pod m naključno izbranimi polji se skrivajo mine. Igralec v vsaki potezi izbere (odpre) eno od še ne odprtih polj. Če polje vsebuje mino, takoj izgubi, v nasprotnem primeru pa se na tistem polju prikaže skupno število min v njegovih neposrednih sosedih. Upoštevajo se tudi diagonalni sosedje, ne samo vodoravni in navpični, zato največje možno število min v soseščini znaša 8. Če je število min v soseščini enako 0, se odprejo vsa sosednja polja. Če kateri od sosedov nima v svoji soseščini nobene mine, se odprejo tudi vsi njegovi sosedje itd. Igralec zmaga, ko so odprta vsa polja razen tistih, ki vsebujejo mino.

Igre žal ni vedno mogoče dobiti brez sreče. Na primer, v situaciji na sliki 1 se mina nahaja bodisi na polju (0, 0) bodisi na polju (0, 1); obe možnosti sta enako verjetni.[2] Vsa ostala temnejša polja vsebujejo mine. Če se preostala mina nahaja na polju (0, 1), igralec pa bo odprl polje (0, 0), bo zmagal.

?	?		2	0	0	0	0	0
1	3		2	0	0	0	0	0
0	1	1	1	0	0	0	0	0
0	0	0	0	0	1	1	2	1
0	0	0	0	0	2		4	
0	0	0	1	1	3		6	
0	0	0	1		2	2		
0	0	0	1	1	1	1	2	2
0	0	0	0	0	0	0	0	0

Slika 1: *Skoraj zaključena plošča. Bela polja so odprta, siva pa zaprta.*

Da bi faktor sreče karseda zmanjšali, bodo vaši programi odigrali zelo veliko število iger, pa tudi pravila bomo nekoliko prilagodili: namesto zmag in porazov bomo šteli odprta polja. Če bo vaš program torej naletel na mino tik pred koncem igre, bo prejel skoraj toliko točk, kot če bi zmagal.

[1] `https://en.wikipedia.org/wiki/Minesweeper_(video_game)`

[2] Polja zapisujemo v obliki (v, s), kjer je v indeks vrstice, s pa indeks stolpca.

3 Vaša naloga

Najprej si izberite ime za svoj program. Ime naj bo sestavljeno iz največ 20 znakov. Omejite se na črke angleške abecede, števke in podčrtaje, sicer pa pustite domišljiji prosto pot.

Nato z Učilnice na svoj računalnik prenesite paket `minolovec.zip` in ga razpakirajte. Ustvarila se bo sledeča hierarhija imenikov in datotek:

```
minolovec
|
+-- javadat.txt
|
+-- src
|   |
|   +-- ogrodje
|   |   |
|   |   +-- Igra.java
|   |   +-- Razpored.java
|   |   +-- Stanje.java
|   |
|   +-- skupno
|   |   |
|   |   +-- Stroj.java
|   |   +-- Polje.java
|   |   +-- Konstante.java
|   |
|   +-- s12345678
|       |
|       +-- Stroj_Nakljucko.java
|
+-- classes
```

V imeniku `minolovec/src` ustvarite podimenik `svvvvvvvv`, pri čemer niz *vvvvvvvv* zamenjajte s svojo vpisno številko. V tem podimeniku izdelajte datoteko `Stroj_`*ime*`.java`, pri čemer niz *ime* nadomestite z imenom svojega programa. Datoteka `Stroj_`*ime*`.java` naj ima takšno vsebino:

```
package svvvvvvvv;

import skupno.*;

public class Stroj_ime implements Stroj {

    @Override
    public void zacetek(int visina, int sirina, int stMin) {
        ...
    }

    @Override
    public Polje izberi(int[][] stanje, long cas) {
        ...
    }
```

```
    @Override
    public void konecIgre(boolean[][] mine, int razlog, int stOdprtih) {
        ...
    }

    // morebitne ostale metode
}
```

V datoteki boste torej definirali razred, ki implementira vmesnik `Stroj` v paketu (podimeniku) `skupno`. Pojasnimo pomen posameznih elementov razreda:

Metoda `zacetek`. Ogrodje (razred `Igra`) bo ob zagonu ustvarilo objekt razreda `Stroj_`*ime*, nato pa bo poklicalo metodo `zacetek`. Njeni parametri po vrsti podajajo višino in širino igralne plošče ter število min. V tej metodi boste najverjetneje inicializirali podatkovne strukture.

Metoda `izberi`. Pred vsako potezo bo ogrodje poklicalo metodo `izberi`. Tabela `stanje` podaja trenutno stanje posameznih polj. Celica `stanje[i][j]` se nanaša na polje v vrstici z indeksom `i` in stolpcu z indeksom `j`. Vrednost v celici je število med -1 in 8: vrednost -1 pove, da je polje še zaprto, vrednosti med 0 in 8 pa podajajo število min v neposredni soseščini polja. Parameter `cas` podaja čas v milisekundah, ki ga ima program na voljo do konca igre.

Metoda `izberi` mora vrniti polje (kot objekt razreda `Polje` v imeniku `skupno`), ki naj ga ogrodje odpre. **Pozor:** če metoda vrne neobstoječe ali že odprto polje, se igra takoj konča (rezultat je enak, kot če bi program odprl polje z mino).

Metoda `konecIgre`. Ogrodje pokliče to metodo ob koncu vsake igre. Parameter `mine` podaja razporeditev min; če se mina nahaja na polju (i, j), potem velja `mine[`i`][`j`] == true` in obratno. Parameter `razlog` podaja razlog za zaključek igre (`Konstante.ZMAGA`, `Konstante.MINA`, `Konstante.NEVELJAVNO_POLJE`, `Konstante.ZE_ODPRTO_POLJE` ali `Konstante.PREKORACITEV_CASA`). Parameter `stOdprtih` podaja število uspešno odprtih polj.

To metodo lahko implementirate s praznim telesom, lahko pa jo uporabite za zbiranje statistike.

Vaša rešitev je lahko sestavljena iz več datotek; edina omejitev je ta, da se ime nobene druge datoteke ne začne z nizom `Stroj_`. Seveda se morajo tudi druge datoteke pričeti s sledečo vrstico:

```
package svvvvvvvv;
```

Na učilnico oddajte paket ZIP (ne RAR ali kaj drugega, prosim!) celotnega podimenika `svvvvvvvv`. (Če paket razpakirate, se mora ustvariti imenik `svvvvvvvv` z vsemi datotekami.)

4 Naključko

Na tekmovanju bo sodeloval tudi stroj Naključko (podimenik `s12345678`). Žal ni pretirano inteligenten, saj — razen v prvi potezi, ko brez izjeme izbere polje $(\lfloor h/2 \rfloor, \lfloor w/2 \rfloor)$ — polja odpira povsem naključno, kljub temu pa pazi, da ne odpre neveljavnega ali že odprtega polja. Zaradi svoje enostavnosti tudi ne more prekoračiti časovne omejitve.

Naključkova koda lahko služi kot izhodišče za vašo.

5 Prevajanje

Program lahko prevedete tako, da se v terminalu postavite v imenik `minolovec` in izvršite sledeči ukaz:

```
javac -d classes @javadat.txt
```

S parametrom `-d` povemo, naj prevajalnik datoteke `.class` shrani v imenik `classes`. (Pri programih, porazdeljenih po več imenikih, je lepše, če so datoteke `.class` ločene od datotek z izvorno kodo.) Parameter `@javadat.txt` prevajalniku naroči, naj prevede vse datoteke, nanizane v datoteki `javadat.txt`. Seveda boste morali tudi svoje datoteke `.java` dodati v datoteko `javadat.txt`.

6 Zagon ogrodja

Ogrodje lahko poženete tako, da se v terminalu prestavite v imenik `minolovec/classes` in izvršite ukaz sledeče oblike:

```
java ogrodje.Igra
    (((1 | 2 | 3 | 4 | hxw_m) [-s seme]) | -r datoteka)
    [Stroj]
    [-t čas]
```

Prvi sklop parametrov podaja velikost plošče, število min in po želji njihovo razporeditev. Ta sklop lahko vsebuje enega od sledečih parametrov:

- `1`: ustvari ploščo velikosti 9×9 z 10 minami (12,3% min);
- `2`: ustvari ploščo velikosti 16×16 s 40 minami (15,6% min);
- `3`: ustvari ploščo velikosti 16×30 z 99 minami (20,6% min);
- `4`: ustvari ploščo velikosti 24×30 s 180 minami (25,0% min);
- *h*`x`*w*`_`*m*: ustvari ploščo velikosti $h \times w$ z m minami;
- `-r` *datoteka*: velikost plošče ter število in razpored min prebere iz podane datoteke. Ta datoteka naj vsebuje h vrstic s po w znaki `+` in `-`, pri čemer znak `+` predstavlja mino, znak `-` pa prazno polje. Na primer, za razporeditev s slike 1 (predpostavimo, da se edina neznana mina nahaja na polju (0, 1)) bi datoteka izgledala takole:

  ```
  -++------
  --+------
  ---------
  ---------
  ------+-+
  ------+-+
  ----+--++
  ---------
  ---------
  ```

Če podamo parameter `1`, `2`, `3` ali `4`, lahko s parametrom `-s` *seme* določimo še seme naključnega generatorja, s pomočjo katerega se razporejajo mine po plošči. Pri istem semenu, isti velikosti plošče in istem številu min bo razporeditev min po plošči vedno enaka.

Pri parametrih `1`, `2`, `3`, `4` in `hxw_m` bo ogrodje mine vedno razporedilo tako, da na polju $(\lfloor h/2 \rfloor, \lfloor w/2 \rfloor)$ in njegovih osmih sosedih ne bo nobene mine. Na ta način bo lahko vaš program v prvi potezi izbral polje $(\lfloor h/2 \rfloor, \lfloor w/2 \rfloor)$ in bo tako imel dobre možnosti za spodobno izhodišče.

Če boste velikost plošče in/ali razporeditev min nastavili sami, upoštevajte omejitve $4 \leqslant h \leqslant 100$, $4 \leqslant w \leqslant 100$ in $1 \leqslant m \leqslant hw - 10$. Če razporeditev podate v datoteki, se lahko mine nahajajo kjerkoli, tudi na sredini plošče.

Parameter *Stroj* podaja razred, ki implementira vmesnik `skupno.Stroj`. Razred podajte v obliki `svvvvvvvv.Stroj_`*ime*. Če ga izpustite, bo ogrodje odigralo igro z vami osebno.

Parameter `-t` *čas* podaja čas v sekundah (ne milisekundah!), ki ga ima stroj na voljo za *celotno* igro. Če tega parametra ne navedete, bo časovna omejitev znašala 5 sekund.

Sledi nekaj primerov zagona ogrodja:

- `java ogrodje.Igra 2 s12345678.Stroj_Nakljucko`

  Stroj `Nakljucko` igra na plošči 16×16 s 40 minami. Za celotno igro ima na voljo 5 sekund.

- `java ogrodje.Igra 20x10_70 -s 12345 s63200999.Stroj_Strucko -t 1`

  Stroj `Strucko` igra na plošči 20×10 s 70 minami. Za celotno igro lahko porabi 1 sekundo časa. Naključni generator, ki ga ogrodje uporabi za razporeditev min po plošči, se inicializira s semenom 12345.

- `java ogrodje.Igra -r mojRazpored.txt s63200999.Stroj_Strucko`

  Stroj `Strucko` igra na plošči, katere velikost in razpored min sta podana v datoteki `mojRazpored.txt`.

7 Dovoljena in nedovoljena sredstva

Kršitev sledečih pravil pomeni diskvalifikacijo, kršitev prvega pa še kaj več:

- **Sleherna vrstica programov mora biti vaša in samo vaša.** Lahko si pomagate s spletnimi viri, nikakor pa ne smete od tam (ali od koderkoli drugod) prepisovati oz. kopirati programske kode. Kršitev tega pravila se bo štela kot goljufanje, to pa ne bo pomenilo samo diskvalifikacije, ampak tudi obravnavo pred fakultetno disciplinsko komisijo (z verjetno enoletno prepovedjo opravljanja izpita).
- Vaš program ne sme ustvarjati procesov ali niti.
- Vaš program ne sme delati z datotečnim sistemom. Bere lahko izključno s standardnega vhoda, piše pa lahko izključno na standardni izhod.
- Vaš program ne sme uporabljati refleksije (paketa `java.lang.reflect`).

8 Potek tekmovanja

Vsak oddani stroj bomo po n-krat pognali z vsakim od sledečih ukazov:

`java ogrodje.Igra 1 -s` $s + 4i$ *Stroj*
`java ogrodje.Igra 2 -s` $s + 4i + 1$ *Stroj*
`java ogrodje.Igra 3 -s` $s + 4i + 2$ *Stroj*
`java ogrodje.Igra 4 -s` $s + 4i + 3$ *Stroj*

Števili n in s se bosta določili po zaključku roka za oddajo. Število n bo odvisno od števila prispelih strojev in dejanskega časa njihovega izvajanja. Število i je indeks iteracije (0: prva; 1: druga; ...; $n - 1$: zadnja).

Število točk za ploščo s parametrom k ($k \in \{1, 2, 3, 4\}$) se bo izračunalo po formuli

$$T = \frac{2^{k-1}o}{hw - m},$$

pri čemer je o število uspešno odprtih polj.

9 Nagrade

Avtor prvouvrščenega programa bo pri predmetu Programiranje 1 neposredno prejel oceno 10 in bo s tem oproščen opravljanja izpita. Avtorji, ki bodo zavzeli mesta od drugega do desetega, pa bodo na izpitu prejeli dodatne točke:

Mesto	Dodatne točke
2.	20
3.	16
4.	12
5.	8
6.–10.	4

Na izpitu bo v skupnem seštevku »prakse« in »teorije« možno zbrati 100 točk, zato se bodo navedene nagrade lahko kar pošteno občutile.

Veliko užitkov pri programiranju!