

Neobligāts papilduzdevums

Izstrādāt programmu, kas meklē ceļu labirintā no ieejas līdz izejai. Labirints tiek aprakstīts ar divdimensiju masīvu no nullēm un vieniniekiem (1 – siena, 0 – ceļš). Ieeja ir pirmais masīva elements (ar indeksiem [0][0]), bet izeja – pēdējais (ar indeksiem [R-1][K-1], kur R ir rindu skaits masīvā, bet K ir kolonu skaits). Paredzēt programmā iespēju automātiski ģenerēt labirintus. Piemēram, labirinta apraksts var izskatīties sekojošā veidā:

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 |

Programmu izstrādāt grupās pa 2-4 studentiem. Studentiem ir jāsadala grupās patstāvīgi (pēc vēlēšanās). Programmā implementēt vairākus ceļa meklēšanas algoritmus (minimums 2 dažādus algoritmus). Labirinta automātiskai ģenerācijai izstrādāt vismaz vienu algoritmu. Izstrādāto programmu augšupielādēt ORTUSā nodemonstrēt mācībspēkam 7. mācību nedēļā.

Visus izstrādātos algoritmus izpētīt: empīriski salīdzināt pēc ātrdarbības; pēc atrastā ceļa garuma, ja ir vairāki ceļi; noskaidrot kā labirinta lielums ietekmē uz algoritma izpildes laiku. Ja ir izstrādāti vairāki labirinta ģenerācijas algoritmi, tad salīdzināt arī tos. Par pētījuma rezultātiem sagatavot pārskatu, kuru ir jāizvieto ORTUS vidē. ORTUS vidē arī ir jāizvieto izstrādāto programmu. Pārskatu par pētījumu un programmu ir jāizvieto tikai vienam no grupas locekļiem. Pārskatam par pētījumu ir jāsaturs informāciju par pētījuma rezultātiem, izstrādātiem algoritmiem un grupas locekļiem (vārdi, uzvārdi, apliecības numuri, grupas numuri).

Trešā mācību nedēļā, visām grupām, kas piedalās dotā uzdevuma risināšanā, ir jāpagatavo un augšupielādē ORTUSā pārskatu, kas satur informāciju par grupu un izvēlēto algoritmu aprakstus. Prasības pārskatam atrodas dotā faila beigās.

Visi grupas locekļi saņems vienu un to pašu bonusa punktu daudzumu (līdz 8 punktiem, atkarībā no pārskatu un pētījuma kvalitātes, no veiksmīgi implementēto algoritmu daudzuma).

Tādējādi, programmai ir jāizpilda sekojošas darbības:

- Ievada no tastatūras divdimensiju masīva izmērus (rindu un kolonu skaitu).
- Ievada no tastatūras divdimensiju masīva elementus vai automātiski aizpilda masīvu ar datiem, atkarība no lietotāja izvēles. Ja masīvs tika aizpildīts ar datiem automātiski, tad izvadīt masīvu ekrānā.
- Izvada implementēto ceļa meklēšanas algoritmu skaitu un ievada no tastatūras algoritma numuru (veselo skaitli).
- Ja ceļš ir, tad izvada ceļu. Ceļu izvadīt sekojošā formātā: (0,0) (1,0) (1,1) (1,2) (1,3) (2,3) ... (6,6).

Programmas struktūras paraugs:

```
package dip107;

import java.util.Scanner;

public class Labirints {

    public static void main(String[] args) {
        int[][] l;
        int[][] path = {{0,0},{0,1},{0,2},{1,2}}; // šeit ir jābūt rezultātam
        int r, c, m;
        char ans;
        Scanner sc = new Scanner(System.in);
        System.out.print("row count: ");
        r = sc.nextInt();

        System.out.print("column count: ");
        c = sc.nextInt();

        sc.nextLine();
        System.out.print("Auto fill maze (y/n)?");
        ans = sc.nextLine().charAt(0);

        if (ans == 'n') {
            l = new int[r][c];
            for (int i=0; i<r; i++)
                for (int j=0; j<c; j++)
                    l[i][j] = sc.nextInt();
        }
        else {
            // automātiski aizpildam masīvu un izvadām to ekrānā
        }

        System.out.print("method number (1-3):");
        m = sc.nextInt();

        switch (m) {
            case 1: //1. algoritms
                break;
            case 2: //2. algoritms
                break;
            case 3: //3. algoritms
                break;
        }

        // rezultātu izvade
        System.out.println("results:");
        for (int i=0; i<4; i++)
            System.out.print("(" + path[i][0] + "," + path[i][1] + ") ");
    }
}
```

Programmas izpildes piemērs:

```
row count: 2
column count: 3
Auto fill maze (y/n)?n
0 0 0
1 1 0
method number (1-3):1
results:
(0,0) (0,1) (0,2) (1,2)
```

Prasības pārskatam, kas satur informāciju par grupu

Sagatavot un augšupielādēt ORTUSā pārskatu, kas satur ziņas par izveidotu grupu un projekta izstrādes plāniem. Pārskatam ir jābūt pdf formātā un jāsaturs sekojošas nodaļas:

I. Grupas apraksts

- Grupas nosaukums.

- Grupas locekļi.

Norādīt vārdus, uzvārdus, grupas numurus un studenta apliecības numurus

- Grupas vadītāja vārds un uzvārds.

Grupas vadītājs ir atbildīgs par pārskatu, pirmkodu augšupielādi ORTUS portālā, kā arī par pienākumu un darba sadali starp grupas locekļiem.

- Problēmu risināšanas metodes.

Aprakstīt kādā veidā grupa plāno risināt radušās darba gaitā problēmas, piemēram, ko grupa darīs, ja viens no locekļiem neizpildīs savus pienākumus: neizstrādās kodu, savlaicīgi neizvietos darba rezultātus ORTUSā un t.t. Tādu problēmu risināšanas ceļi var būt dažādi, piemēram, palikušie grupas biedri var pieņemt lēmumu izpildīt visu darbu, vai izslēgt no grupas studentu, kas nestrādā.

II. Komunikācijas metodes un izvēlētās tehnoloģijas

- Komunikācijas metodes.

Aprakstīt, kā notiks komunikācija starp grupas locekļiem un cik bieži grupas locekļiem ir jāsazinās ar grupas vadītāju un savā starpā, lai atskaitītos par darba progresu (telefonu zvani, SMS, e-pasta vēstules, Viber, WhatsApp...; katru dienu vai vienu reizi nedēļā...).

- Izstrādes vides un versiju kontroles sistēmas (replit, GitHub.com, Bitbucket.org).

III. Algoritmu apraksts

Projekta ietvaros katrai grupai ir jāizstrādā programmu, programmu, kas meklē ceļu labirintā no ieejas līdz izejai. Programmā ir jārealizē minimums 2 algoritmus. Aprakstīt pārskatā vispārējā veidā izvēlētos algoritmus izmantojot dabīgo valodu, pseidokodu, vai grafisko algoritma shēmu.

IV. Programmas struktūras apraksts

Sastādīt tabulas, kas apraksta klases un metodes, kuras plānots izstrādāt. Tabulas var izskatīties sekojošā veidā.

| | |
|------------------|---------|
| Class1 | |
| Metodes apraksts | Method1 |
| Metodes apraksts | Method2 |
| | |

V. Pienākumu sadale starp grupas locekļiem

Sadaliet pienākumus, aizpildiet sekojošu tabulu, norādot, kurš no grupas locekļiem izstrādās kādu no klasēm un metodēm. Novērtējiet darba apjomu, vajadzīgo katra uzdevuma izpildīšanai (no 1 līdz 30 punktiem). Cenšaties, lai noslodze grupā būtu sadalīta vienmērīgi.

| # | Uzdevums | Darba apjoms | Izpildītājs | | |
|----|----------------------------|-----------------|-------------|-------|------|
| | | | Māris | Jānis | Ilze |
| 1. | Class1.method1 | 20 | | 20 | |
| 2. | Class1.method2 | 10 | | | 10 |
| 3. | Class2.method1 | 15 | 15 | | |
| 4. | Prezentācijas sagatavošana | 10 | | | 10 |
| 5. | Testa piemēru sagatavošana | 20 | 10 | 5 | 5 |
| | ... | | | | |
| | Kopā: | | 25 | 25 | 25 |