

I. GRUPAS APRAKSTS

-Grupas nosaukums.

POGramētāji

-Grupas locekļi.

- Tomass Kristiāns Šterns, 5.grupa, 231RDB190
- Petr Gabuniia, 6.grupa, 231RDB331
- Valentīns Koposovs, 11.grupa, 231RDB008

-Grupas vadītāja vārds un uzvārds.

Tomass Kristiāns Šterns

-Problēmu risināšanas metodes.

Ja kāds no grupas locekļiem neizpildīs savus pienākumus pārskata izveidē, notiks pārrunas un, iespējams, tiks nolemts, vai šim cilvēkam būs jāuzņemas kādi papilduzdevumi un/vai prezentācijas veidošana. Ja kāds no grupas locekļiem neizpildīs savus pienākumus programmas (*.java) izveidē un/vai prezentācijas izveidē, tiks nolemts, vai šim cilvēkam būs jāuzņemas kādi papilduzdevumi un/vai prezentēšana. Ja grupas vadītājs neiesniedz nepieciešamos failus un/vai nepilda savus pienākumus, tiks nolemts, vai grupas vadītājam tiks atņemts viņa statuss. Kā arī katra grupas locekļa pienākumu nepildīšana var tikt uzskatīts kā iemesls viņa atskaitīšanai no grupas.

II. KOMUNIKĀCIJAS METODES

Komunikācija notiek klātienē, ja tas ir iespējams, vai sazinoties grupas Whatsapp sarakstē. Atskaites notiek Whatsapp-ā, github-ā un klātienē, aprakstot izdarīto.

III. ALGORITMU APRAKSTS

1. Brute force – programma ies cauri labirintam soli pa solim līdz neatradīs izeju vai līdz nebeidzas neizietie lauciņi. Tā nemeklēs obligāti īsāko ceļu, bet pirmo ceļu, kuru atrod gājienu prioritātē – uz leju, pa labi, pa kreisi, uz augšu. Ja programma pamana, ka tā atkal atgriežas lauciņā, kurā bija, tad atzīmē, ka tur vēlreiz iet nevajag, tādā veidā izejot visu iespējamo labirintu vai līdz brīdim, kad atrod izeju pirmo reizi.
2. Grafu metode – programma sastāda grafu, kas attēlo labirintu, kur virsotnes reprezentē ceļus (rūtiņas, pa kuram var iet), bet virsotnes tiek savienotas ar

šķautni gadījumā ja no vienas virsotnes var aiziet uz otru ar vienu gājieni. Šī metode reducē uzdevumu masīva uz uzdevumu grafā, ar kuriem ir ērtāk strādāt. Ceļa noteikšana izmantots Dziļuma pirmās meklēšanas (Depth-First Search) algoritms, kurs nodrošina rekursīvu pārvietošanos pa grafa virsotnēm, meklējot ceļu.

3. Follow right hand rule – algoritms “izstaigās” labirintu, ejot gar labo malu jeb “turot roku” pie labās malas. Algoritms noris pēc noteiktas struktūras: eksistē kāds virziens, kurā aplūko nākamo lauciņu. Ja pa labi attiecībā pret šo virzienu ir ceļš, tad pēc labās rokas likuma pagriežas pa labi. Ja tāds ceļš neeksistē, tad turpina virzību taisni. Ja tas nav iespējams, nogriežas pa kreisi un turpina ceļu. Taču ja neatrod nevienu ceļu, tad ir nonākts strupceļā, tāpēc apgriežas apkārt un atkārtō šo procesu, kamēr atrod izeju. Ja izeja nav atrasta, tas nozīmē, ka algoritms ir atgriezies ieejas pozīcijā 1-2 reizes (atkarībā no labirinta izkārtojuma) un algoritma darbība tiek pārtraukta. Šāds process funkcionē, jo labirintam ir garantēta ieeja un izeja, tās ir tikai vienas un visas ārējās sienas, kas neietilpst labirintā, ir savienotas.

IV. PROGRAMMAS STRUKTŪRAS APRAKSTS

Main	public class. Galvenā klase.
Main.printLabyrinth	Izveda labirintu uz ekrāna. Kā mainīgo pieņem labirinta masīva atsauci.
Main.randomAizpilde	Aizpilda labirintu, izmantojot Math.random(), ar 0 vai 1. Kā mainīgo pieņem labirinta masīva atsauci.
Main.bruteForce	Iziet visus ceļus labirintā soli pa solim līdz brīdim, kad atrod izeju pirmo reizi, vai līdz beidzas lauciņi, kur iet. Kā mainīgo pieņem labirinta masīva atsauci. Sīkāka info “III. ALGORITMU APRAKSTS”
Main.followRightHandRule	Seko labās puses sienai, kamēr atrod izeju, vai atgriežas ieejas pozīcijās noteiktu reižu skaitu. Kā mainīgo pieņem labirinta masīva atsauci. Sīkāka info “III. ALGORITMU APRAKSTS”
Main.createEdges	Grafā savieno ar šķautnēm virsotnēs, kuram no vienas uz otru var aiziet ar vienu gājieni, izveido un izveda ceļu
Main.dfs	Funkcija, kas implementē Dziļuma pirmās meklēšanas (Depth-First Search) algoritmu. Tas ir rekursīvs algoritms, ar

	kura palīdzība tika īstenota pārvietošana pa grafu virsotnēm un ceļa meklēšana līdz izejai.
LabyrinthGraph	Klase, kas ļauj attēlot labirintu kā grafu, kurā virsotnes reprezentē ceļus (rūtiņas, pa kurām var iet), bet virsotnes tiek savienotas ar šķautni gadījumā ja no vienas virsotnes var aiziet uz otru ar vienu gājieni
LabyrinthGraph.addEdge	Savieno 2 virsotnes ar šķautnēm
LabyrinthGraph.VertexInfo	Izveda informāciju par grafu virsotnēm

V. PIENĀKUMU SADALE STARP GRUPAS LOCEKĻIEM

#	Uzdevums	Darba apjoms	Izpildītājs		
			Tomass	Petr	Valentīns
1.	Main.printLabyrinth	3		2	
2.	Main.randomAizpilde	3		3	
3.	Main.bruteForce	30			30
4.	Main.graph	30		30	
5.	Main.followRightHandRule	25	25		
6.	Prezentācijas sagatavošana	30	10	10	10
7.	Testa piemēru sagatavošana	27	9	9	9
8.	Pārskata izveidošana	30	15	5	10
	Kopā:		59	59	59