

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

ФАКУЛЬТЕТ ЭЛЕКТРОННО-ИНФОРМАЦИОННЫХ СИСТЕМ

Кафедра интеллектуальных информационных технологий

Отчет по лабораторной работе №7

Специальность ПО5

Выполнил:
А.А. Игнатюк,
студент группы ПО-5

Проверил:
А.А. Крощенко,
ст. преп. кафедры ИИТ,
«__» _____ 2021 г.

Брест 2021

Цель работы: Освоить возможности языка программирования Java в построении графических приложений.

Вариант 5.

Задание 1.

Требования к выполнению:

- Реализовать соответствующие классы, указанные в задании.
- Организовать ввод параметров для создания объектов (можно использовать файлы).
- Осуществить визуализацию графических примитивов, решить поставленную задачу.

5) Изобразить в окне приложения (апплета) отрезок, вращающийся в плоскости экрана вокруг одной из своих концевых точек. Цвет прямой должен изменяться при переходе от одного положения к другому.

Спецификация ввода:

“Enter window width: ” <ширина окна приложения>

“Enter window height: ” <высота окна приложения>

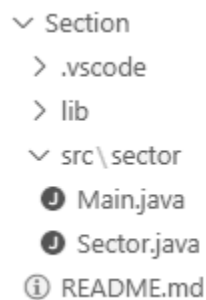
“Enter sector length: ” <длина отрезка>

Спецификация вывода:

<апплет с графическими объектами>

...

Структура проекта:



```
Section
├── .vscode
├── lib
└── src\sector
    ├── Main.java
    ├── Sector.java
    └── README.md
```

Рисунок 1.1 - Структура проекта.

Код программы:

1 Sector.java X

Java > Section > src > sector > 1 Sector.java > ...

```

1  package sector;
2
3  import java.awt.*;
4  import javax.swing.*;
5  import java.lang.Math;
6
7  public final class Sector extends JFrame {
8      private final Integer m_WindowWidth, m_WindowHeight, m_SectorLength, m_StartX, m_StartY;
9      private Integer m_EndX = 0, m_EndY = 0, m_Angle = 0;
10
11     public Sector(final Integer c_WindowWidth, final Integer c_WindowHeight, final Integer c_SectorLength) {
12         super("Sector");
13         this.m_WindowWidth = c_WindowWidth;
14         this.m_WindowHeight = c_WindowHeight;
15         this.m_SectorLength = c_SectorLength;
16
17         this.m_StartX = this.m_WindowWidth / 2;
18         this.m_StartY = this.m_WindowHeight / 2;
19
20         this.m_EndX = this.m_StartX + this.m_SectorLength;
21         this.m_EndY = this.m_StartY;
22
23         this.setVisible(Boolean.TRUE);
24         this.setResizable(Boolean.FALSE);
25         this.setDefaultCloseOperation(EXIT_ON_CLOSE);
26         this.setSize(this.m_WindowWidth, this.m_WindowHeight);
27     }
28
29     public final void paint(final Graphics c_Graphics) {
30         c_Graphics.setColor(Color.WHITE);
31         c_Graphics.fillRect(0, 0, this.m_WindowWidth, this.m_WindowHeight);
32
33         if (this.m_Angle < 90) {
34             c_Graphics.setColor(Color.RED);
35         } else if (this.m_Angle < 180) {
36             c_Graphics.setColor(Color.BLUE);
37         } else if (this.m_Angle < 270) {
38             c_Graphics.setColor(Color.GREEN);
39         } else {
40             c_Graphics.setColor(Color.YELLOW);
41         }
42
43         c_Graphics.drawLine(this.m_StartX, this.m_StartY, this.m_EndX, this.m_EndY);
44     }
45
46     public final void f_run() {
47         while (Boolean.TRUE) {
48             this.repaint();
49
50             m_Angle %= 360;
51             ++this.m_Angle;
52
53             final Double c_AngleInRadians = Math.toRadians(m_Angle),
54                 c_NewEndX = this.m_StartX + (this.m_SectorLength * Math.cos(c_AngleInRadians)),
55                 c_NewEndY = this.m_StartY - (this.m_SectorLength * Math.sin(c_AngleInRadians));
56
57             this.m_EndX = c_NewEndX.intValue();
58             this.m_EndY = c_NewEndY.intValue();
59
60             try {
61                 final Integer c_Sllep = 8;
62                 Thread.sleep(c_Sllep);
63             } catch (final InterruptedException c_Exception) {
64                 System.err.println(c_Exception.getMessage());
65             }
66         }
67     }
68 }
69

```

Рисунок 1.2 - Содержимое файла Sector.java.

```

Main.java X
Java > Section > src > sector > Main.java > ...
1  package sector;
2
3  import java.util.Scanner;
4
5  public final class Main {
    Run | Debug
6      public final static void main(final String[] c_Args) throws Exception {
7          final Scanner c_Scanner = new Scanner(System.in);
8
9          System.out.print("Enter window width: ");
10         final Integer c_WindowWidth = c_Scanner.nextInt();
11
12         System.out.print("Enter window height: ");
13         final Integer c_WindowHeight = c_Scanner.nextInt();
14
15         System.out.print("Enter sector length: ");
16         final Integer c_SectorLength = c_Scanner.nextInt();
17
18         c_Scanner.close();
19
20         Sector v_SectorApp = new Sector(c_WindowWidth, c_WindowHeight, c_SectorLength);
21         v_SectorApp.f_run();
22     }
23 }
24

```

Рисунок 1.3 - Содержимое файла Main.java.

```

PS C:\Users\User\Documents\Visual Studio Code> c:; cd 'c:\Users\User\Documents\Visual Studio Code'; & 'c:\Users\User\.vscode\extensions\vscjava.vscode-java-debug-0.36.0\scripts\launcher.bat' 'C:\Program Files\Eclipse Foundation\jdk-11.0.12.7-hotspot\bin\java.exe' '-Dfile.encoding=UTF-8' '-cp' 'C:\Users\User\AppData\Roaming\Code\User\workspaceStorage\35645f84fe309ce1d5ab2a9af314d21a\redhat.java\jdt_ws\Visual Studio Code_9125e193\bin' 'sector.Main'
Enter window width: 480
Enter window height: 320
Enter sector length: 100

```

Рисунок 1.4 - Запуск программы.



Рисунок 1.5 - Результат работы программы.

Задание 2.

Требования к выполнению:

- Реализовать построение заданного типа фрактала по варианту.
- Везде, где это необходимо, предусмотреть ввод параметров, влияющих на внешний вид фрактала.

5) Дерево Пифагора.

Спецификация ввода:

- “Enter window width: ” <ширина окна приложения>
- “Enter window height: ” <высота окна приложения>
- “Enter tree root length: ” <длина первого отрезка>
- “Enter recursion depth: ” <глубина рекурсии (длина наименьшего отрезка)>
- “Enter tree length K: ” <коэффициент уменьшения длины отрезка>
- “Enter tree right fall: ” <делитель для расчета наклона правой части дерева>
- “Enter tree left fall: ” <делитель для расчета наклона левой части дерева>

Спецификация вывода:

<апплет с графическими объектами>

...

Структура проекта:

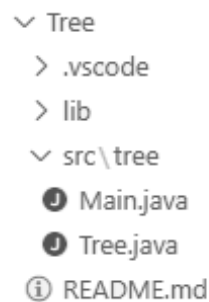


Рисунок 2.1 - Структура проекта.

Код программы:

Main.java X

Java > Tree > src > tree > Main.java > ...

```
1  package tree;
2
3  import java.util.Scanner;
4
5  public final class Main {
    Run | Debug
6      public final static void main(final String[] c_Args) throws Exception {
7          final Scanner c_Scanner = new Scanner(System.in);
8
9          System.out.print("Enter window width: ");
10         final Integer c_WindowWidth = c_Scanner.nextInt();
11
12         System.out.print("Enter window height: ");
13         final Integer c_WindowHeight = c_Scanner.nextInt();
14
15         System.out.print("Enter tree root length: ");
16         final Integer c_RootLength = c_Scanner.nextInt();
17
18         System.out.print("Enter recursion depth: ");
19         final Integer c_Depth = c_Scanner.nextInt();
20
21         System.out.print("Enter tree length K: ");
22         final Double c_LengthK = c_Scanner.nextDouble();
23
24         System.out.print("Enter tree right fall: ");
25         final Double c_RightFall = c_Scanner.nextDouble();
26
27         System.out.print("Enter tree left fall: ");
28         final Double c_LeftFall = c_Scanner.nextDouble();
29
30         c_Scanner.close();
31
32         Tree v_TreeApp = new Tree(c_WindowWidth, c_WindowHeight, c_RootLength, c_Depth, c_LengthK, c_RightFall,
33         |         c_LeftFall);
34         v_TreeApp.f_run();
35     }
36 }
37
```

Рисунок 2.2 - Содержимое файла Main.java.

Java > Tree > src > tree > Tree.java > ...

```

1  package tree;
2
3  import javax.swing.*;
4  import java.awt.*;
5
6  class Tree extends JFrame {
7      private final Integer m_WindowWidth, m_WindowHeight, m_RootLength, m_Depth;
8      private final Double m_LengthK, m_RightFall, m_LeftFall;
9
10     public Tree(final Integer c_WindowWidth, final Integer c_WindowHeight, final Integer c_RootLength,
11                final Integer c_Depth, final Double c_LengthK, final Double c_RightFall, final Double c_LeftFall) {
12         this.m_WindowWidth = c_WindowWidth;
13         this.m_WindowHeight = c_WindowHeight;
14         this.m_RootLength = c_RootLength;
15         this.m_Depth = c_Depth;
16
17         this.m_LengthK = c_LengthK;
18         this.m_RightFall = c_RightFall;
19         this.m_LeftFall = c_LeftFall;
20
21         this.setVisible(Boolean.TRUE);
22         this.setResizable(Boolean.FALSE);
23         this.setDefaultCloseOperation(EXIT_ON_CLOSE);
24         this.setSize(this.m_WindowWidth, this.m_WindowHeight);
25         this.setBackground(Color.WHITE);
26     }
27
28     public final void f_run() {
29         while (Boolean.TRUE) {
30             this.repaint();
31
32             try {
33                 final Integer c_Sllep = 8;
34                 Thread.sleep(c_Sllep);
35             } catch (final InterruptedException c_Exception) {
36                 System.err.println(c_Exception.getMessage());
37             }
38         }
39     }
40
41     public final void paint(final Graphics c_Graphics) {
42         this.f_draw(m_WindowWidth / 2, m_WindowHeight, this.m_RootLength, Math.PI / 2.0, c_Graphics);
43     }
44
45     private final void f_draw(final Integer c_StartX, final Integer c_StartY, final Integer c_Length,
46                              final Double c_Alpha, final Graphics c_Graphics) {
47         int v_Length = c_Length;
48
49         if (v_Length > this.m_Depth) {
50             v_Length = (int) (v_Length * this.m_LengthK);
51
52             final Double c_EndX = c_StartX + v_Length * Math.cos(c_Alpha),
53                     c_EndY = c_StartY - v_Length * Math.sin(c_Alpha);
54
55             c_Graphics.drawLine(c_StartX, c_StartY, c_EndX.intValue(), c_EndY.intValue());
56
57             this.f_draw(c_EndX.intValue(), c_EndY.intValue(), v_Length, c_Alpha + Math.PI / this.m_RightFall,
58                        c_Graphics);
59             this.f_draw(c_EndX.intValue(), c_EndY.intValue(), v_Length, c_Alpha - Math.PI / this.m_LeftFall,
60                        c_Graphics);
61         }
62     }
63 }
64

```

Рисунок 2.3 - Содержимое файла Tree.java.

```

PS C:\Users\User\Documents\Visual Studio Code> c:; cd 'c:\Users\User\Documents\Visual Studio Code'; & 'c:\User
s\User\.vscode\extensions\vscjava.vscode-java-debug-0.36.0\scripts\launcher.bat' 'C:\Program Files\Eclipse Foun
dation\jdk-11.0.12.7-hotspot\bin\java.exe' '-Dfile.encoding=UTF-8' '-cp' 'C:\Users\User\AppData\Roaming\Code\Us
er\workspaceStorage\35645f84fe309ce1d5ab2a9af314d21a\redhat.java\jdt_ws\Visual Studio Code_9125e193\bin' 'tree.
Main'
Enter window width: 1920
Enter window height: 1080
Enter tree root length: 400
Enter recursion depth: 2
Enter tree length K: 0.7
Enter tree right fall: 5.0
Enter tree left fall: 5.0

```

Рисунок 2.4 - Запуск программы.

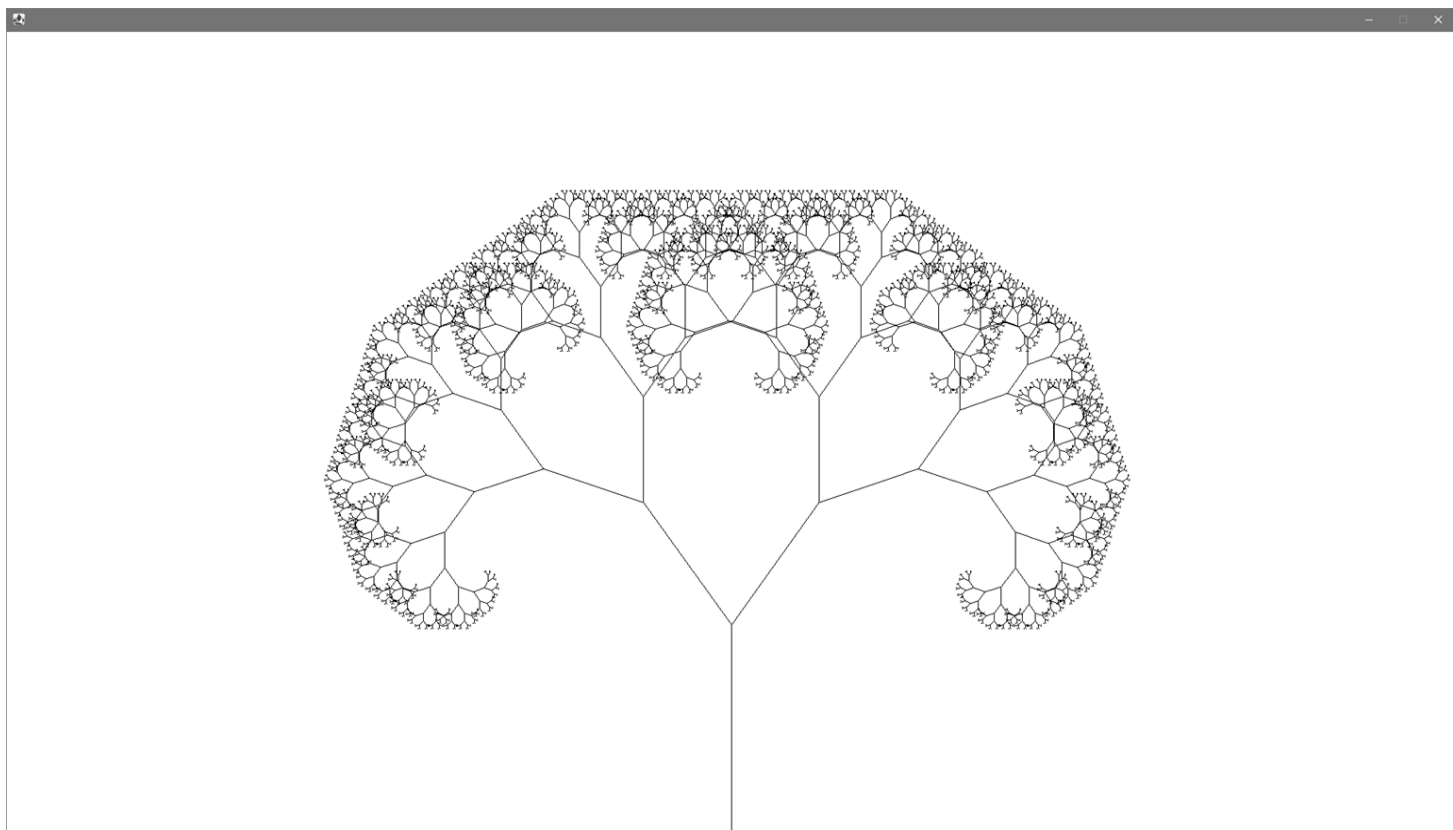


Рисунок 2.5 - Результат работы программы.

```

PS C:\Users\User\Documents\Visual Studio Code> c:; cd 'c:\Users\User\Documents\Visual Studio Code'; & 'c:\User
s\User\.vscode\extensions\vscjava.vscode-java-debug-0.36.0\scripts\launcher.bat' 'C:\Program Files\Eclipse Foun
dation\jdk-11.0.12.7-hotspot\bin\java.exe' '-Dfile.encoding=UTF-8' '-cp' 'C:\Users\User\AppData\Roaming\Code\Us
er\workspaceStorage\35645f84fe309ce1d5ab2a9af314d21a\redhat.java\jdt_ws\Visual Studio Code_9125e193\bin' 'tree.
Main'
Enter window width: 1920
Enter window height: 1080
Enter tree root length: 200
Enter recursion depth: 4
Enter tree length K: 0.8
Enter tree right fall: 7.0
Enter tree left fall: 7.0

```

Рисунок 2.6 - Запуск программы.

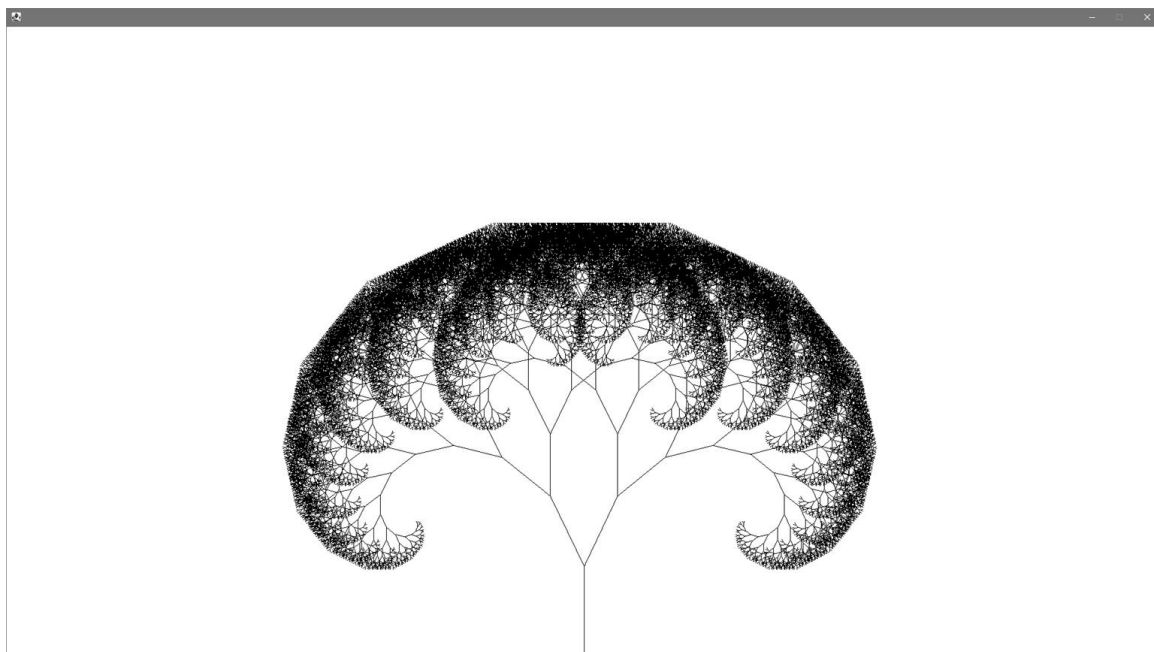


Рисунок 2.7 - Результат работы программы.

```
PS C:\Users\User\Documents\Visual Studio Code> c:; cd 'c:\Users\User\Documents\Visual Studio Code'; & 'c:\Users\User\vscode\extensions\vscjava.vscode-java-debug-0.36.0\scripts\launcher.bat' 'C:\Program Files\Eclipse Foundation\jdk-11.0.12.7-hotspot\bin\java.exe' '-Dfile.encoding=UTF-8' '-cp' 'C:\Users\User\AppData\Roaming\Code\User\workspaceStorage\35645f84fe309ce1d5ab2a9af314d21a\redhat.java\jdt_ws\Visual Studio Code_9125e193\bin' 'tree.Main'
Enter window width: 1920
Enter window height: 1080
Enter tree root length: 300
Enter recursion depth: 10
Enter tree length K: 0.75
Enter tree right fall: 8.0
Enter tree left fall: 5.0
□
```

Рисунок 2.8 - Запуск программы.

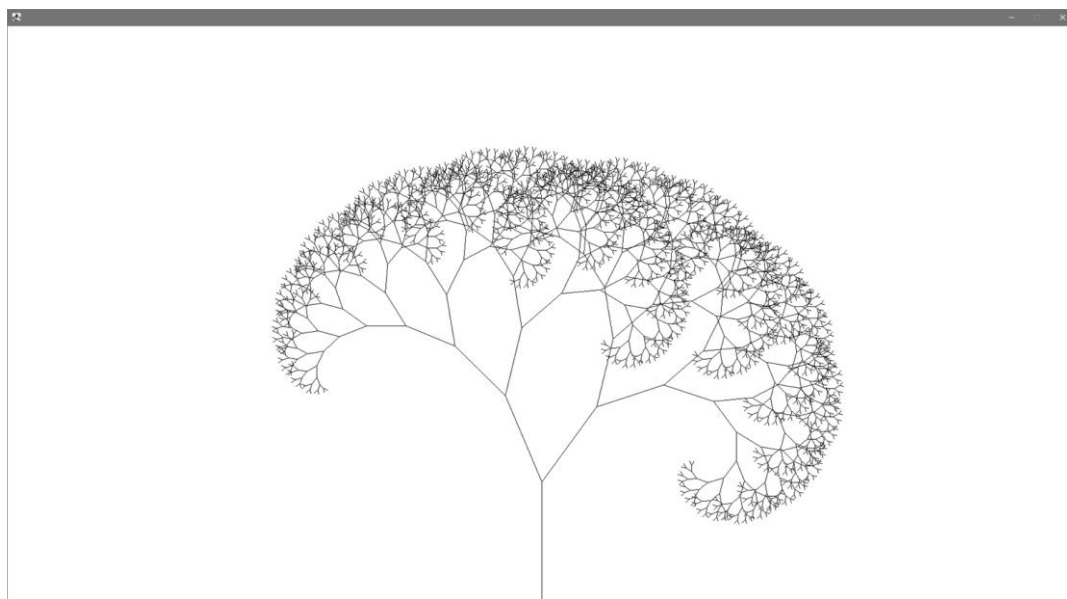


Рисунок 2.9 - Результат работы программы.

Вывод: Освоил возможности языка программирования Java в построении графических приложений.