

Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и кибербезопасности
Высшая школа «Компьютерных технологий и информационных систем»

ОТЧЕТ

по дисциплине «Системный подход в разработке программного обеспечения»

Выполнил:

студент группы 5130902/20201

_____ А. И. Сафонов
подпись

Проверил:

Кандидат тех. Наук., доцент

_____ С. А. Нестеров
подпись

«__» _____ 2024г.

Санкт-Петербург, 2024

Оглавление

1. Смотрим версии Java Puzzle Ball	3
16.08.2013	3
22.08.2013	4
27.09.2013	5
16.10.2013	5
21.11.2013	6
2. Упражнения из занятия 2-2 Задачи программы	8
Упражнение 1.	8
Упражнение 2.	12
Упражнение 3.	13
3. Упражнения – Раздел 2	17
Проблема 1: Простая ASCII-графика	17
Проблема 2: Исходная ASCII-графика.....	17
Проблема 3: The Snake Box Factory	18

1. Смотрим версии Java Puzzle Ball

16.08.2013

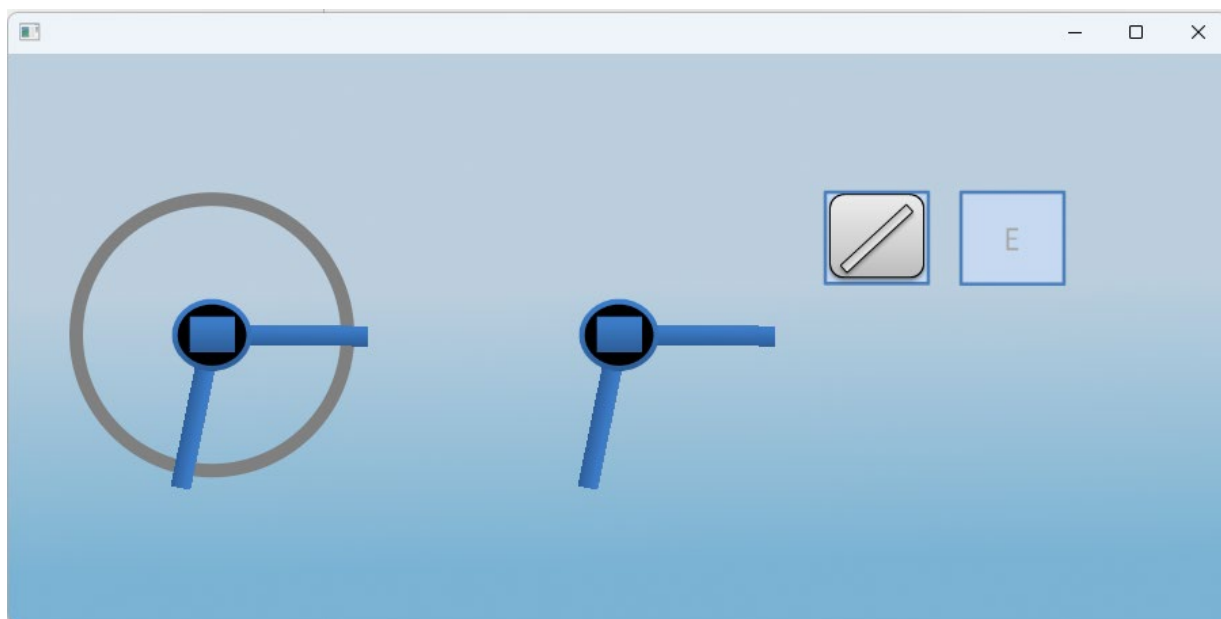


Рисунок 1.1 – Версия игры Java Puzzle Ball 08-16-13

Функции:

- Отображение графических объектов на экране.
- Распознавание событий мыши.
- Вращение объектов BlueVumper.
- Перетаскивание значка в области (N, E).

22.08.2013

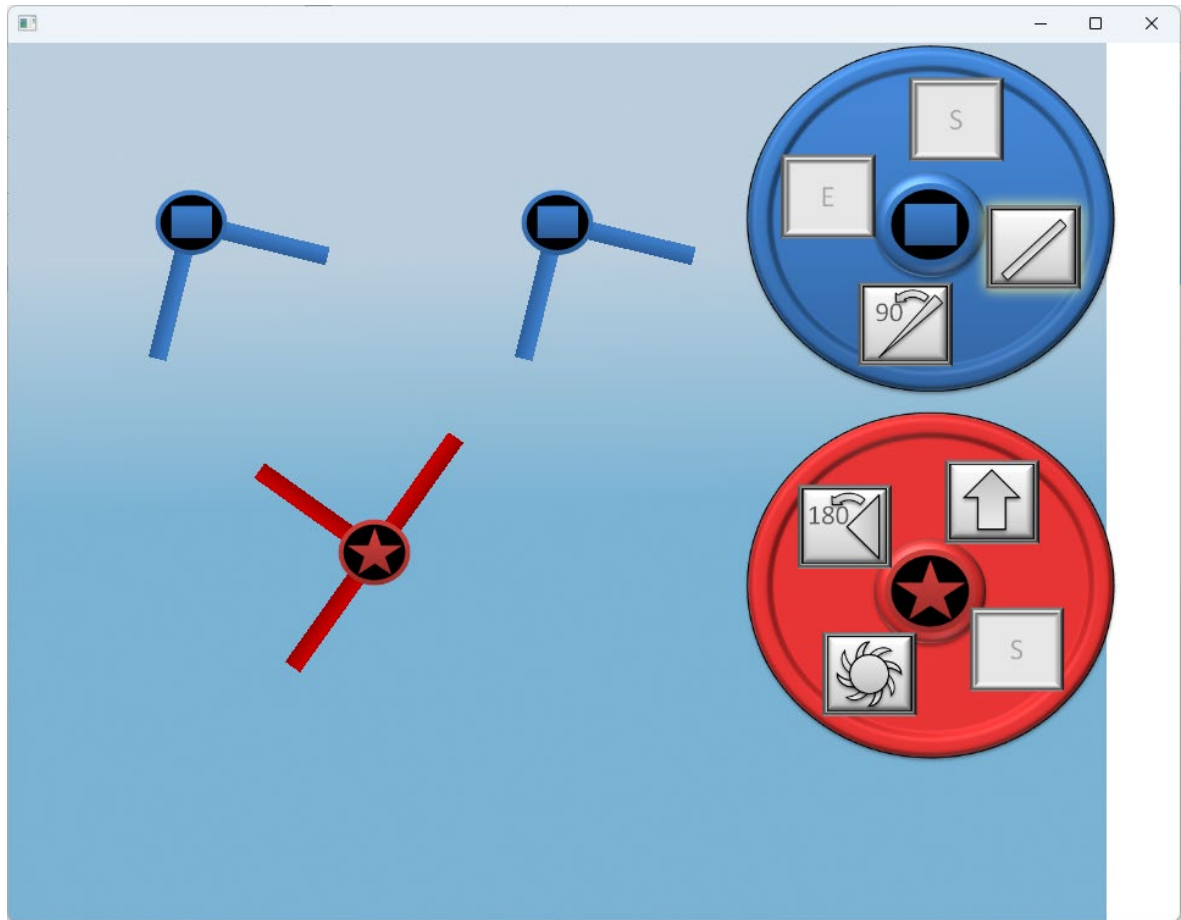


Рисунок 1.2 – Версия игры Java Puzzle Ball 09-16-13

Функции:

- Элементы пользовательского интерфейса (UI).
- Колесики и значки — расположены справа.
- Объект RedBumper.
- Цветные вложения.
- Дополнительные значки для перетаскивания.

27.09.2013

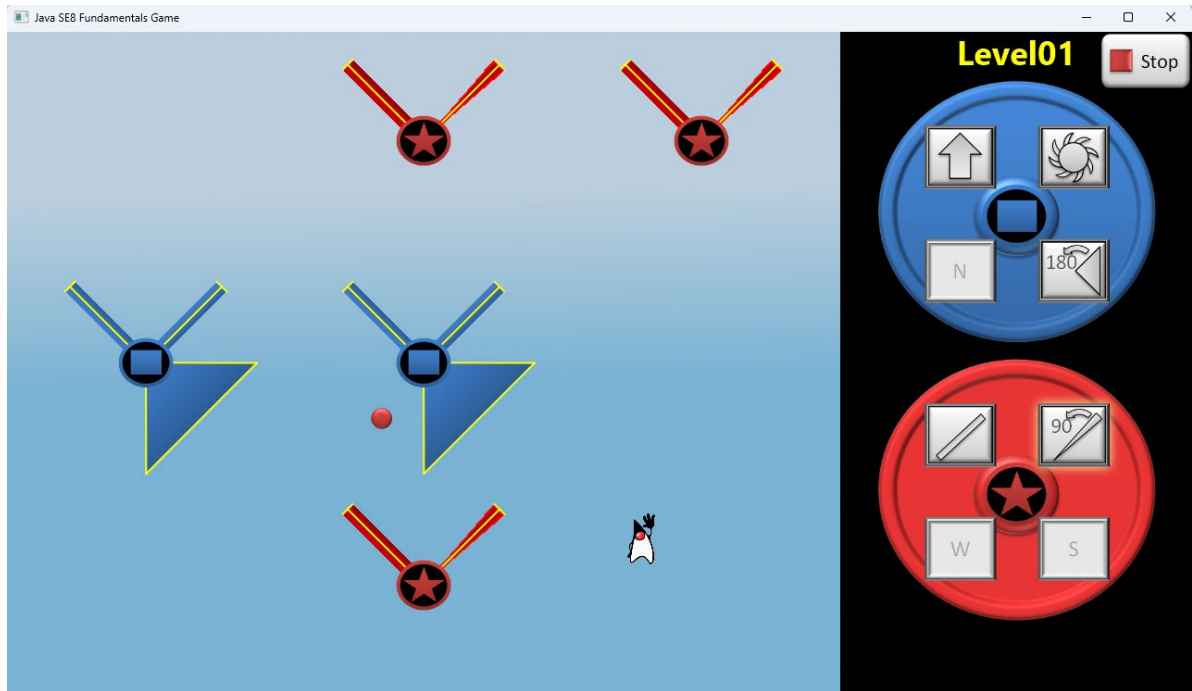


Рисунок 1.3 – Версия игры Java Puzzle Ball 09-27-13

Задача — перенаправить мяч Дюку.

Функции:

- Кнопка Play и цель (Дюк)
- Возможность менять направление движущегося мяча
- Возможность прикреплять дополнительные фигуры
- Желтые линии (для распознавания столкновений)
- Колесики, которые занимают ближайшее положение с шагом 45 градусов

16.10.2013

Появилось всплывающее меню для выбора уровней.



Рисунок 1.4 – Всплывающее меню для выбора уровней

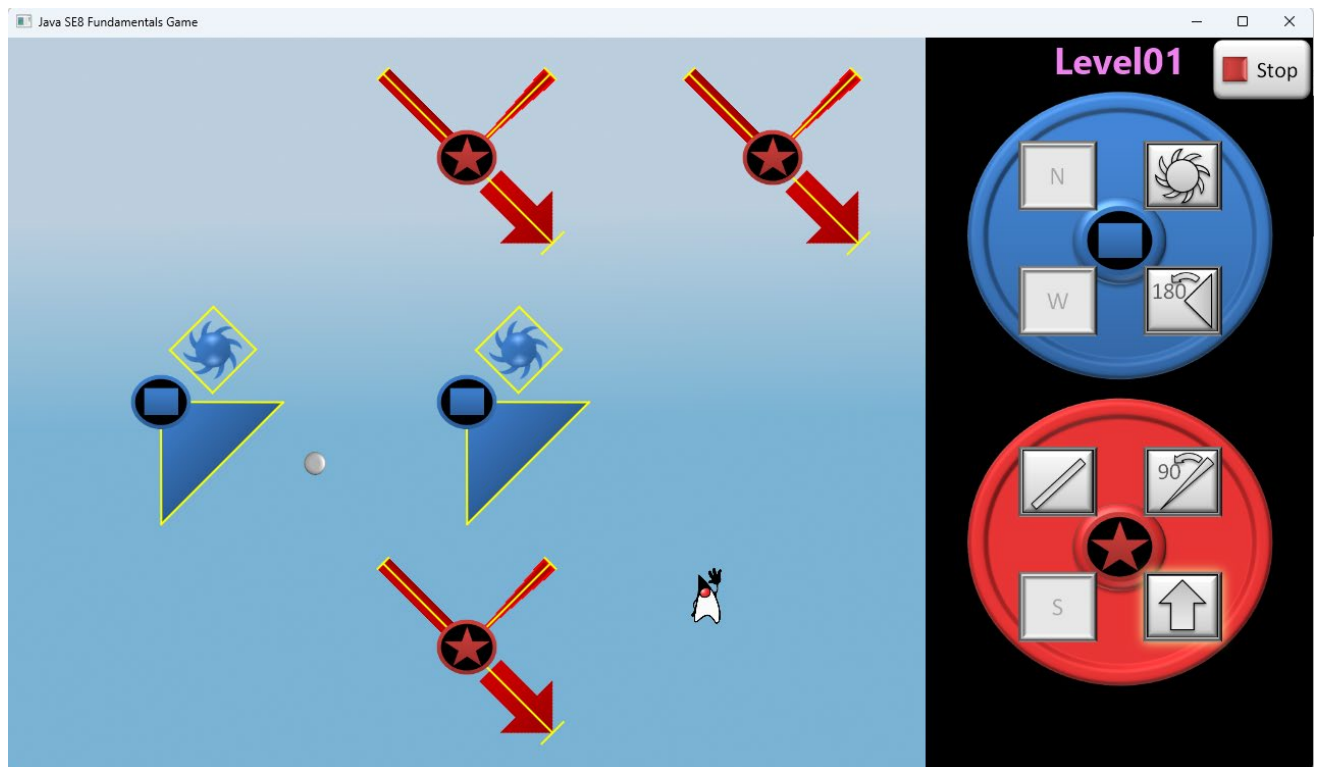


Рисунок 1.5 – Версия игры Java Puzzle Ball 10-16-13

Функции:

- Геометрия уровня
- Инструкции по созданию уровней считываются из текстового файла

21.11.2013

Уровней стало больше, теперь выбор уровня реализован в виде всплывающего списка.

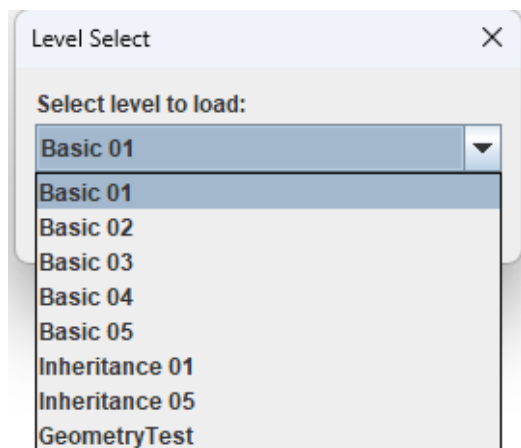


Рисунок 1.6 – Всплывающее меню для выбора уровней в виде всплывающего списка

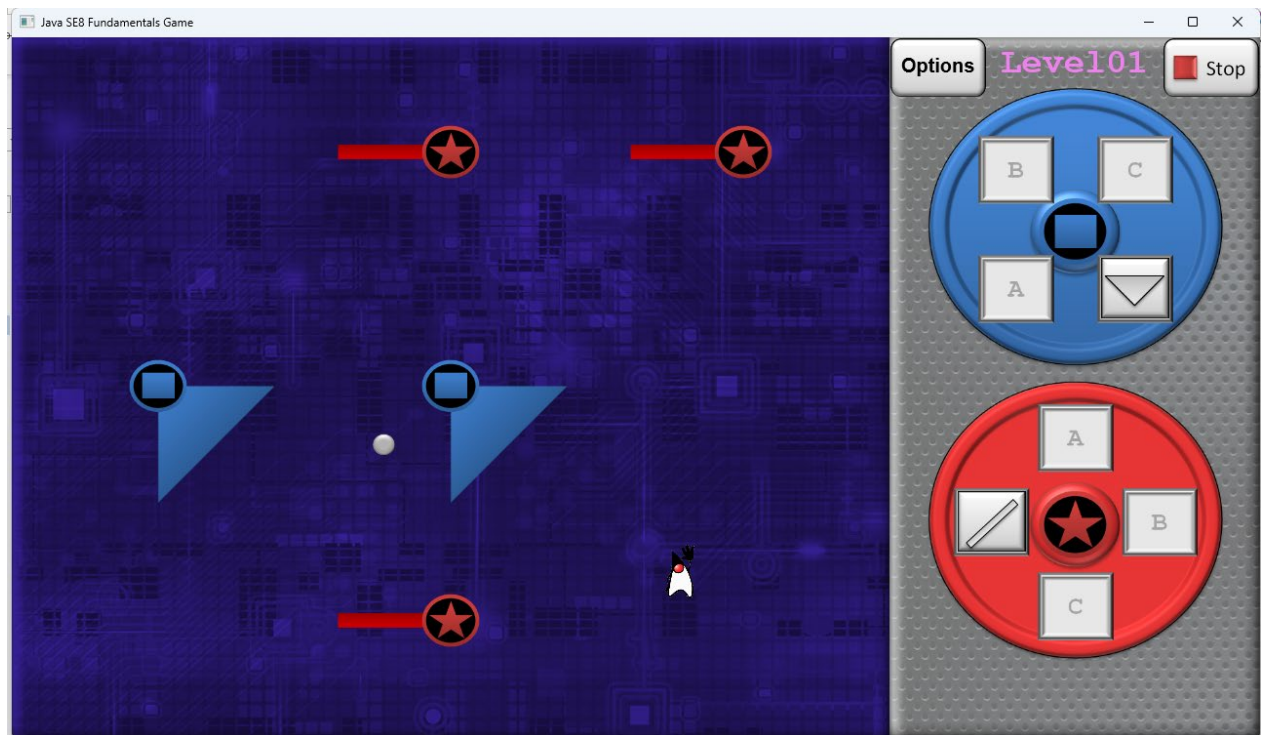


Рисунок 1.7 – Версия игры Java Puzzle Ball 11-21-13

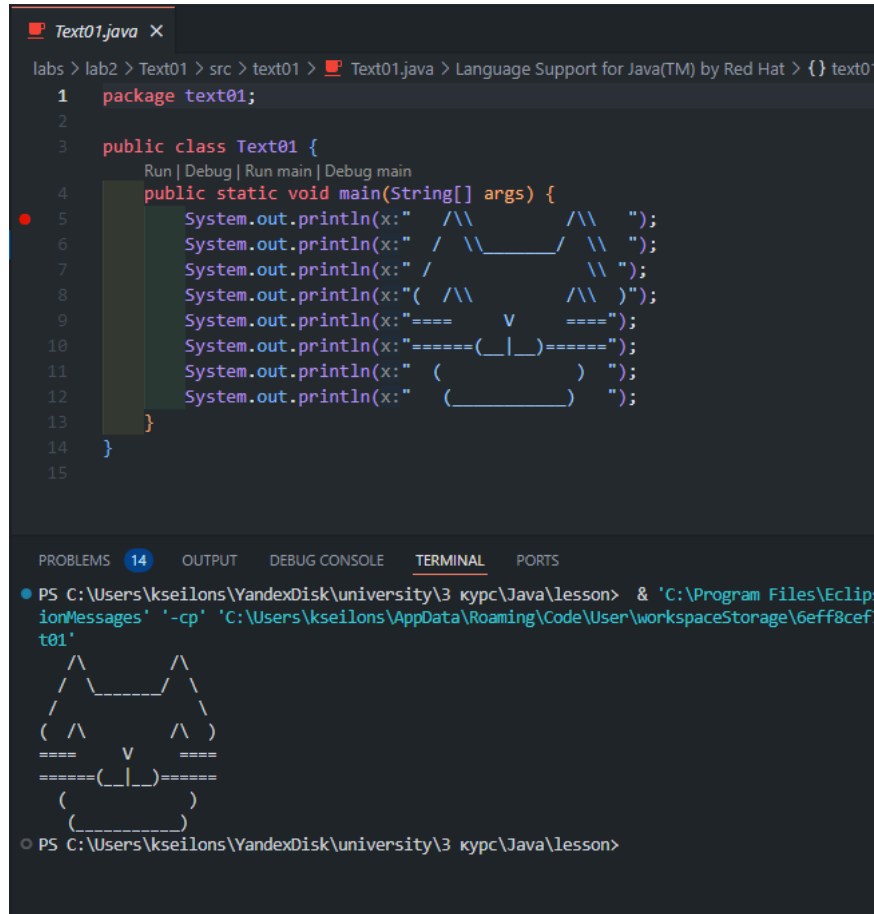
Функции:

- Уровни можно выбирать с помощью кнопки Options
- Новые красивые фоновые изображения
- Дополнительные уровни
- Области теперь называются ABCD вместо NESW

2. Упражнения из занятия 2-2 Задачи программы

Упражнение 1.

Установите точку останова в строке 5. Запустите программу в обычном режиме.



The screenshot shows an IDE window titled 'Text01.java'. The code is as follows:

```
1 package text01;
2
3 public class Text01 {
4     public static void main(String[] args) {
5         System.out.println(x:"  /\      /\  ");
6         System.out.println(x:" /  \____/  \ ");
7         System.out.println(x:" /              \ ");
8         System.out.println(x:"(  /\      /\  )");
9         System.out.println(x:"====  v  =====");
10        System.out.println(x:"=====(_|_)=====");
11        System.out.println(x:" (              ) ");
12        System.out.println(x:" (_____) ");
13    }
14 }
15
```

Below the code editor, the 'TERMINAL' tab is active, showing the command prompt output:

```
PS C:\Users\kseilons\YandexDisk\university\3 курс\Java\lesson> & 'C:\Program Files\Eclipse
ionMessages' '-cp' 'C:\Users\kseilons\AppData\Roaming\Code\User\workspaceStorage\6eff8cef
t01'
  /\      /\
 /  \____/  \
 /              \
(  /\      /\  )
====  v  =====
=====(_|_)=====
 (              )
 (_____)
PS C:\Users\kseilons\YandexDisk\university\3 курс\Java\lesson>
```

Рисунок 2.1– Запуск программы в обычном режиме

Запустите программу с отладчиком.

- Должно появиться окно "Output".
- Нажмите "Step Over", чтобы перейти на следующую строку.


```
labs > lab2 > Text01 > src > text01 > Text01.java > ...
1 package text01;
2
3 public class Text01 {
4     Run | Debug | Run main | Debug main
5     public static void main(String[] args) { args = String[0]@10
6         System.out.println(x:"  /\ \      /\ \  ");
7         System.out.println(x:" /  \ \_____/  \ \  ");
8         System.out.println(x:" /                    \ \  ");
9         System.out.println(x:"(  /\ \      /\ \  )");
10        System.out.println(x:"====      V      =====");
11        System.out.println(x:"=====(_|_)=====");
12        System.out.println(x:" (                    ) ");
13        System.out.println(x:"  (_____)  ");
14    }
15 }
```

Рисунок 2.2– Запуск программы в debug mode до нажатия на “ Step Over ”

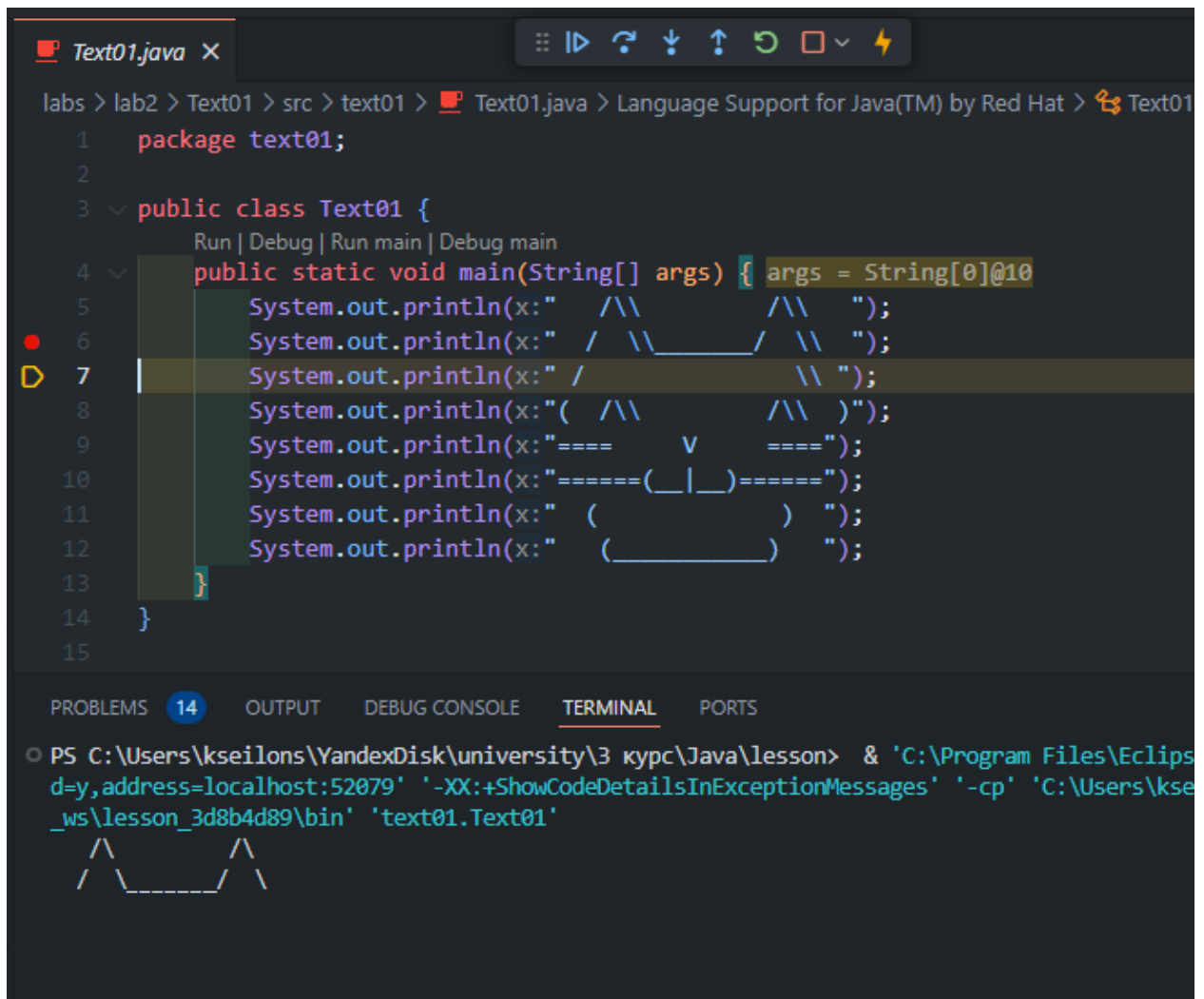


Рисунок 2.3– Запуск программы в debug mode после нажатия на “ Step Over ”

- Измените код, чтобы первые три оператора print отображались в строке 5. (Это называется "удаление пробелов".)

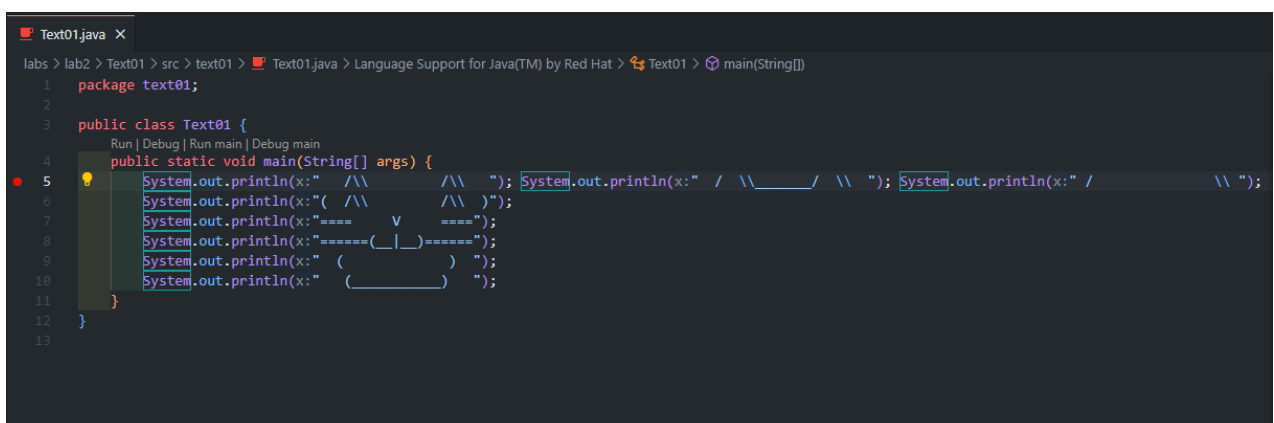


Рисунок 2.4– Изменение кода: отступы первых трех операторов print в строке 5 убраны

- Запустите программу с отладчиком.
 - Должно появиться окно "Output".

- Нажимайте кнопку "Step Over Expression", чтобы перейти к каждой следующей строке.

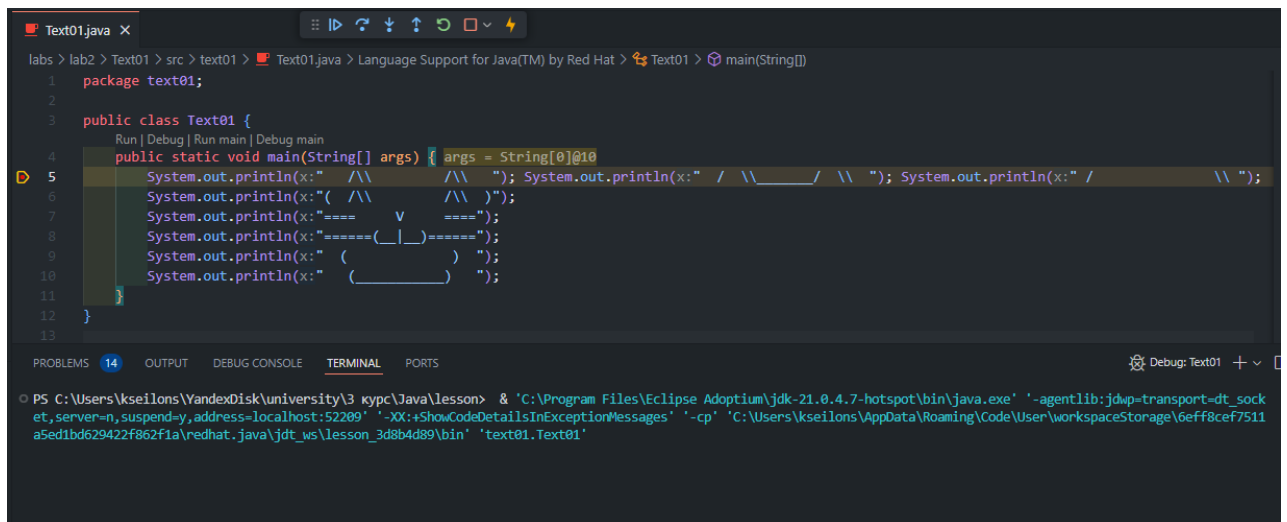


Рисунок 2.5— Запуск программы в debug mode перед нажатием на “ Step Out ”

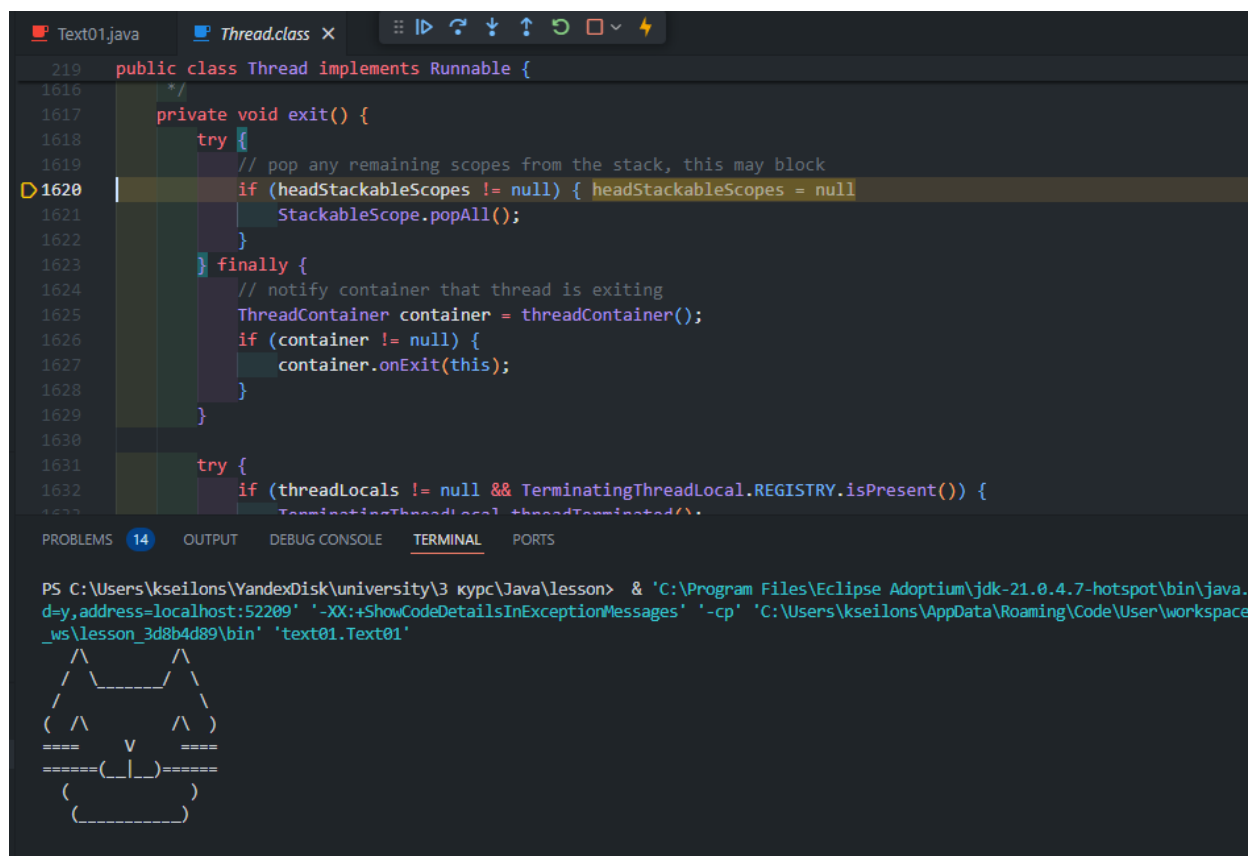


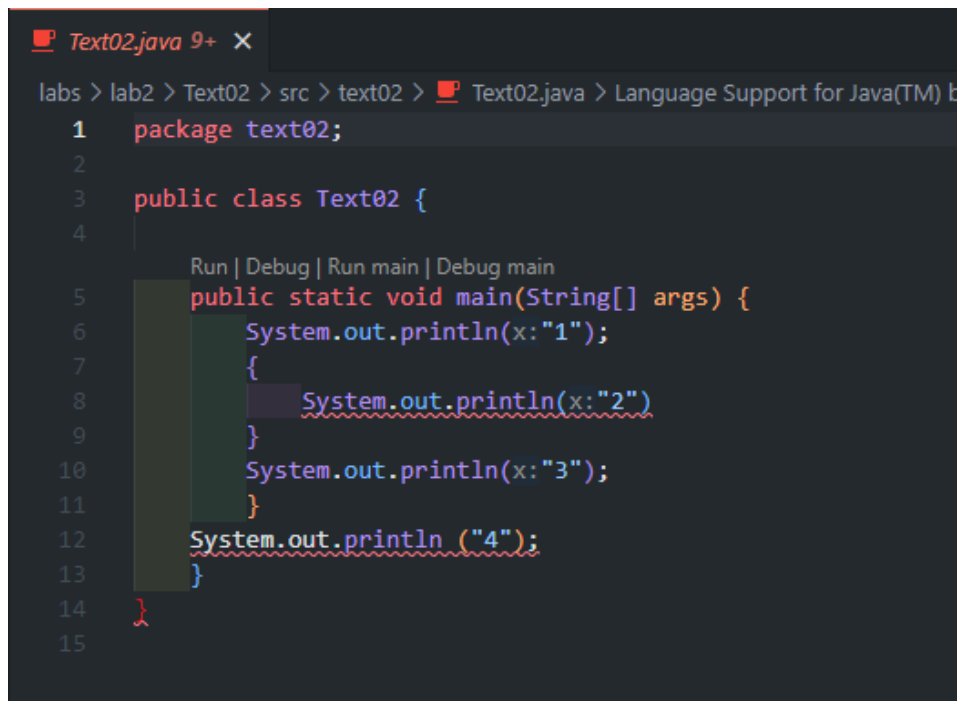
Рисунок 2.6— Запуск программы в debug mode после нажатия на “ Step Out ”

- Наблюдайте за тем, как появляется изображение кошки.
- Попробуйте удалить точку с запятой во время отладки программы.

Если удалить точку с запятой во время отладки программы ничего не произойдет, это не повлияет на текущую отладку. Чтобы изменения вступили в силу, нужно перекомпилировать программу. Если мы удалим точку с запятой перед запуском, программа не скомпилируется.

Упражнение 2.

Попробуйте исправить эту программу:

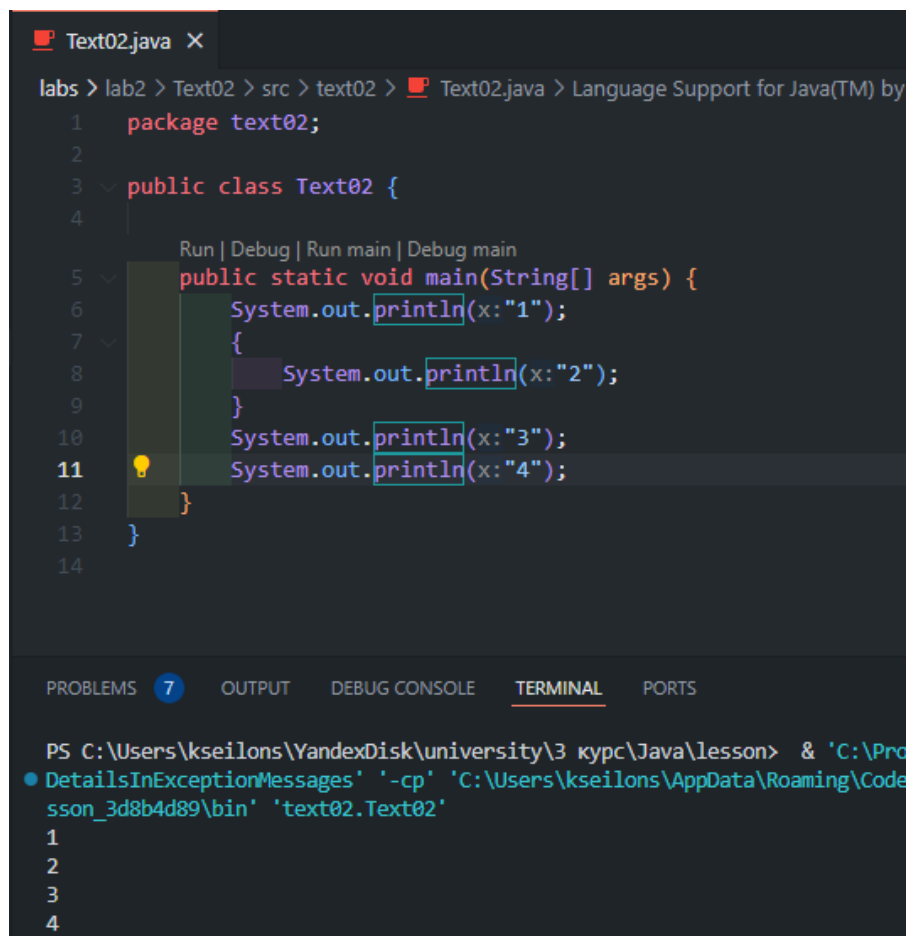


The screenshot shows an IDE window titled 'Text02.java 9+ X'. The code is as follows:

```
1 package text02;
2
3 public class Text02 {
4
5     Run | Debug | Run main | Debug main
6     public static void main(String[] args) {
7         System.out.println(x:"1");
8         {
9             System.out.println(x:"2")
10        }
11        System.out.println(x:"3");
12    }
13    System.out.println ("4");
14 }
15
```

The code contains several syntax errors: a stray 'x:' in the first print statement, an opening curly brace on line 8 without a closing one, and a closing curly brace on line 12 without an opening one. The IDE highlights these errors with red squiggly lines.

Рисунок 2.7– Неисправная программа из проекта Text02



The screenshot shows the same IDE window after corrections. The code is now valid:

```
1 package text02;
2
3 public class Text02 {
4
5     Run | Debug | Run main | Debug main
6     public static void main(String[] args) {
7         System.out.println(x:"1");
8         {
9             System.out.println(x:"2");
10        }
11        System.out.println(x:"3");
12        System.out.println(x:"4");
13    }
14 }
```

The IDE shows a lightbulb icon on line 11, indicating a suggestion. The bottom panel shows the 'TERMINAL' tab with the following command and output:

```
PS C:\Users\kseilons\YandexDisk\university\3 курс\Java\lesson> & 'C:\Pro
DetailsInExceptionMessages' '-cp' 'C:\Users\kseilons\AppData\Roaming\Code
sson_3d8b4d89\bin' 'text02.Text02'
1
2
3
4
```

Рисунок 2.8– Исправленная программа из проекта Text02

Упражнение 3.

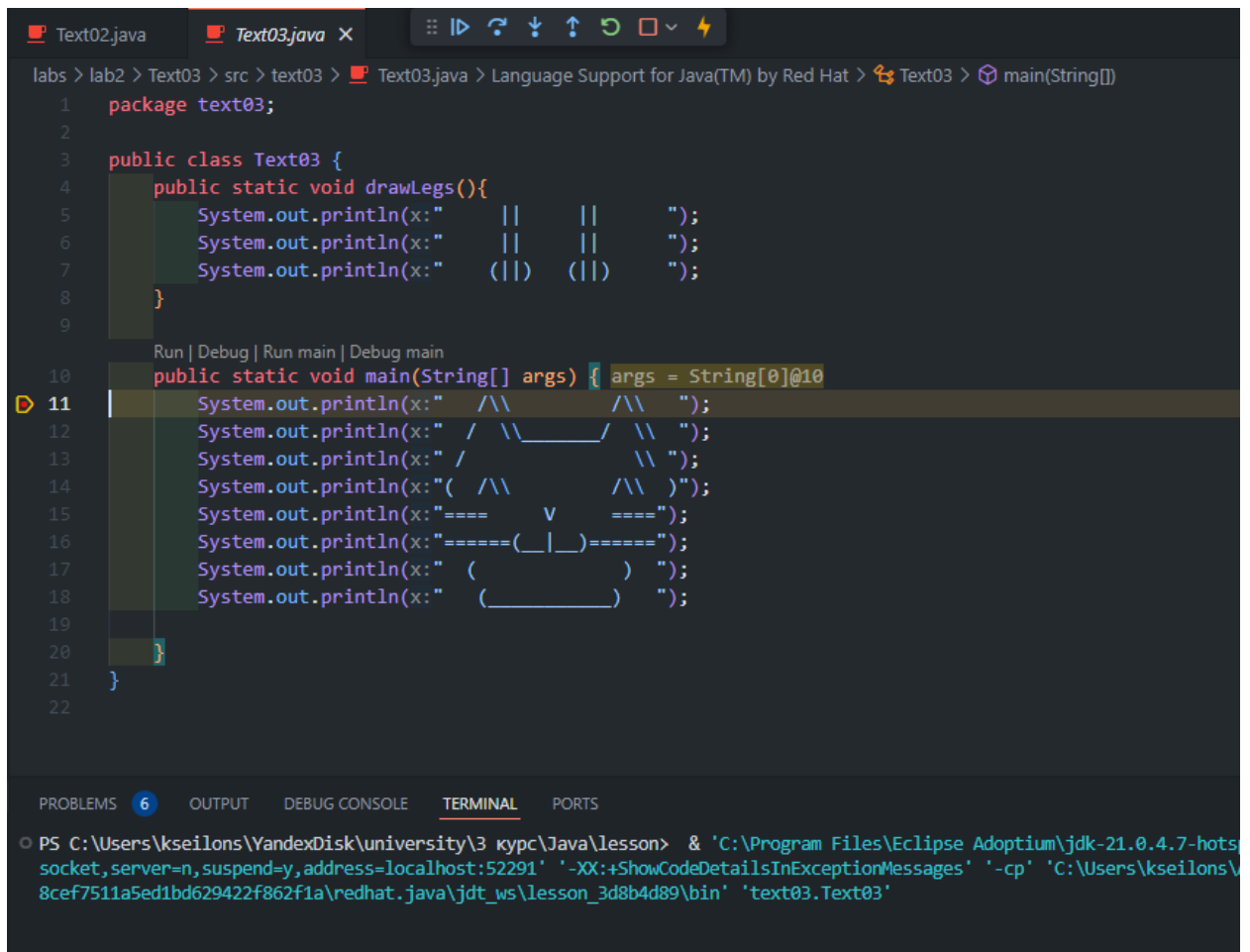
Установите точку останова в строке 11.

Запустите программу с отладчиком.

- Убедитесь, что выбрано окно "Output".
- Нажмите "Step Over", чтобы перейти на следующую строку

Введите drawLegs(); в строке 19 и запустите программу в режиме отладки.

- Где можно добавить точку останова, чтобы увидеть, как рисуются отдельные линии лап?
- Какой получится результат, если закомментировать строки?



```
1 package text03;
2
3 public class Text03 {
4     public static void drawLegs(){
5         System.out.println(x:"    ||    ||    ");
6         System.out.println(x:"    ||    ||    ");
7         System.out.println(x:" (||) (||)    ");
8     }
9
10    Run | Debug | Run main | Debug main
11    public static void main(String[] args) { args = String[0]@10
12        System.out.println(x:"    /\    /\    ");
13        System.out.println(x:" /  \_____/  \    ");
14        System.out.println(x:" /                \    ");
15        System.out.println(x:"(  /\                /\  )");
16        System.out.println(x:"===== v =====");
17        System.out.println(x:"=====(_|_)=====");
18        System.out.println(x:" (                ) ");
19        System.out.println(x:" (                ) ");
20    }
21 }
22
```

PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\kseilons\YandexDisk\university\3 курс\Java\lesson> & 'C:\Program Files\Eclipse Adoptium\jdk-21.0.4-hotspot\socket,server=n,suspend=y,address=localhost:52291' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\kseilons\8cef7511a5ed1bd629422f862f1a\redhat.java\jdt_ws\lesson_3d8b4d89\bin' 'text03.Text03'

Рисунок 2.9– Запуск программы Text03

```
labs > lab2 > Text03 > src > text03 > Text03.java > Language Support for Java(TM) by Red Hat > Text03 > main(String[])
1 package text03;
2
3 public class Text03 {
4     public static void drawLegs(){
5         System.out.println(x:"    ||    ");
6         System.out.println(x:"    ||    ");
7         System.out.println(x:" (||) (||) ");
8     }
9
10    Run | Debug | Run main | Debug main
11    public static void main(String[] args) { args = String[0]@10
12        System.out.println(x:"  /\      /\  ");
13        System.out.println(x:" /  \____/  \ ");
14        System.out.println(x:" /              \ ");
15        System.out.println(x:"(  /\      /\  )");
16        System.out.println(x:"====      v      ===");
17        System.out.println(x:"=====(_|_)=====");
18        System.out.println(x:" (              ) ");
19        System.out.println(x:" (_____) ");
20    }
21 }
22
```

PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\kseilons\YandexDisk\university\3 кырп\Java\lesson> & 'C:\Program Files\Eclipse Adoptium\jdk-21.0.4-hotspot\bin\java.exe socket,server=n,suspend=y,address=localhost:52291' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\kseilons\AppData\Roaming\8cef7511a5ed1bd629422f862f1a\redhat.java\jdt_ws\lesson_3d8b4d89\bin' 'text03.Text03'

/\ /\

Рисунок 2.10— Запуск программы Text03 нажали на кнопку “Step over”

Чтобы увидеть, как рисуются лапы, точку останова нужно установить в пятой строчке:

```

1 package text03;
2
3 public class Text03 {
4     public static void drawLegs(){
5         System.out.println(x:"  ||  ||  ");
6         System.out.println(x:"  ||  ||  ");
7         System.out.println(x:" (||) (||) ");
8     }
9
10    Run | Debug | Run main | Debug main
11    public static void main(String[] args) {
12        System.out.println(x:"  /\\" );
13        System.out.println(x:" /  \_\_\_\_ /  \\" );
14        System.out.println(x:" /              \\" );
15        System.out.println(x:"(  /\\"      /\\" )");
16        System.out.println(x:"====      v      =====");
17        System.out.println(x:"=====(_|_)=====");
18        System.out.println(x:" (              ) ");
19        System.out.println(x:" (_____) ");
20        drawLegs();
21    }
22 }

```

```

PS C:\Users\kseilons\YandexDisk\university\3 kypc\Java\lesson> & 'C:\Program Files\Eclipse Adoptium\jdk-21.0.4.7-hotspot\bin\j
● socket,server=n,suspend=y,address=localhost:52291' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\kseilons\AppData\Ro
8cef7511a5ed1bd629422f862f1a\redhat.java\jdt_ws\lesson_3d8b4d89\bin' 'text03.Text03'
    /\
    /\
PS C:\Users\kseilons\YandexDisk\university\3 kypc\Java\lesson> ^C
PS C:\Users\kseilons\YandexDisk\university\3 kypc\Java\lesson>
PS C:\Users\kseilons\YandexDisk\university\3 kypc\Java\lesson> c:: cd 'c:\Users\kseilons\YandexDisk\university\3 kypc\Java\les
tium\jdk-21.0.4.7-hotspot\bin\java.exe' '-agentlib:jdwp=transport=dt_socket,server=n,suspend=y,address=localhost:52303' '-XX:+S
p' 'C:\Users\kseilons\AppData\Roaming\Code\User\workspaceStorage\6eff8cef7511a5ed1bd629422f862f1a\redhat.java\jdt_ws\lesson_3d8
    /\
    /\
 ( /\      /\ )
====      v      =====
=====(_|_)=====
 (              )
 (_____)

```

Рисунок 2.11– Запуск программы Text03 точка останова на лапах кота

Если закомментировать строки с выводом мордочки кота, например, то она будет игнорироваться:

The screenshot shows the Eclipse IDE with two tabs: Text02.java and Text03.java. The Text03.java tab is active, displaying the following code:

```
1 package text03;
2
3 public class Text03 {
4     public static void drawLegs(){
5         System.out.println(x:"    ||    ||    ");
6         System.out.println(x:"    ||    ||    ");
7         System.out.println(x:"    (||)  (||)  ");
8     }
9
10    Run | Debug | Run main | Debug main
11    public static void main(String[] args) {
12        // System.out.println("    /\\"      /\\"    ");
13        // System.out.println(" /  \\\_\_\_\_\_\_ /  \\" );
14        // System.out.println(" /                      \\" );
15        // System.out.println("(  /\\"      /\\"  )");
16        // System.out.println("====      V      ===");
17        // System.out.println("=====(_|_)=====");
18        // System.out.println("      (      )  ");
19        // System.out.println("      (_____)  ");
20        drawLegs();
21    }
```

The terminal output shows the execution of the program, displaying the following text:

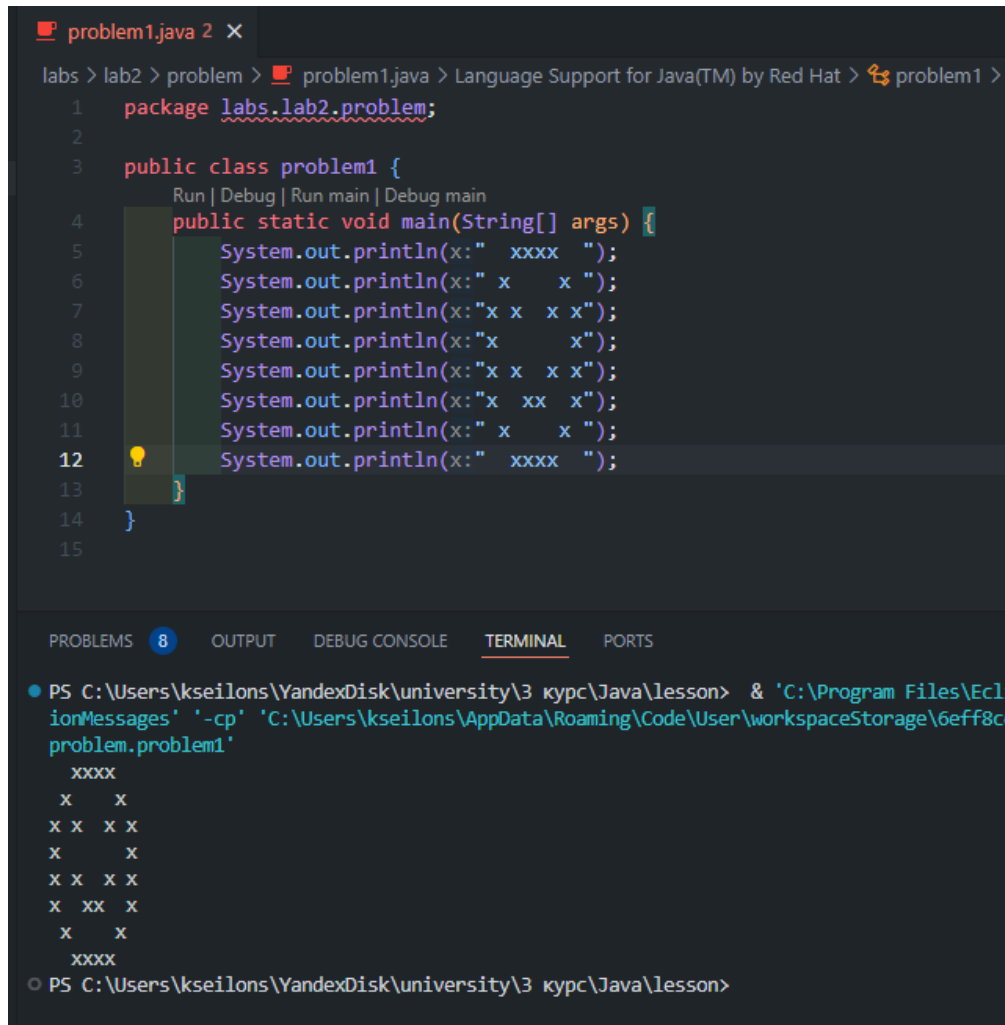
```
PS C:\Users\kseilons\YandexDisk\university\3 кypc\Java\lesson> & 'C:\Program Files\Eclipse Adopt
DetailsInExceptionMessages' '-cp' 'C:\Users\kseilons\AppData\Roaming\Code\User\workspaceStorage\6
sson_3d8b4d89\bin' 'text03.Text03'
    ||    ||
    ||    ||
    (||)  (||)
PS C:\Users\kseilons\YandexDisk\university\3 кypc\Java\lesson>
```

Рисунок 2.12– Запуск программы Text03 кот без мордочки

3. Упражнения – Раздел 2

Проблема 1: Простая ASCII-графика

Воспроизвести приведенное выше улыбающееся лицо, используя 8 операторов вывода. Помимо пробела ваш рисунок будет основан на одном символе, таком как X или #.



The screenshot shows an IDE window titled 'problem1.java 2'. The code is as follows:

```
1 package labs.lab2.problem;  
2  
3 public class problem1 {  
4     public static void main(String[] args) {  
5         System.out.println(x:" xxxx ");  
6         System.out.println(x:" x  x ");  
7         System.out.println(x:"x x  x x");  
8         System.out.println(x:"x    x");  
9         System.out.println(x:"x x  x x");  
10        System.out.println(x:"x  xx x");  
11        System.out.println(x:" x  x ");  
12        System.out.println(x:" xxxx ");  
13    }  
14 }  
15
```

Below the code editor, the 'TERMINAL' tab is active, showing the command to run the program and its output:

```
PS C:\Users\kseilons\YandexDisk\university\3 курс\Java\lesson> & 'C:\Program Files\Eclipse Software\bin\org.eclipse.equinox.launcher_2.0.0.v20180605-1700.jar' -cp 'C:\Users\kseilons\AppData\Roaming\Code\User\workspaceStorage\6eff8c6e\problem.problem1'
```

```
xxxx  
x  x  
x x x x  
x    x  
x x x x  
x  xx x  
x  x  
xxxx
```

The output is a simple ASCII art of a smiling face, where 'x' represents the eyes, nose, and mouth, and spaces represent the cheeks and forehead.

Рисунок 3.1– Улыбающийся смайлик

Проблема 2: Исходная ASCII-графика

Создать собственную оригинальную ASCII-графику, используя операторы вывода. С помощью комментариев опишите, что показано на изображении.

```

labs > lab2 > problem > problem2.java > Language Support for Java(TM) by Red Hat > problem2
1  package labs.lab2.problem;
2
3  public class problem2 {
4      Run | Debug | Run main | Debug main
5      public static void main(String[] args) {
6          System.out.println(x:"      /\\"      ");
7          System.out.println(x:"      /  \\"      ");
8          System.out.println(x:"      /____\\      ");
9          System.out.println(x:"      /|    |\\      ");
10         System.out.println(x:"      /_|___|\\      ");
11         System.out.println(x:"      |_____|      ");
12         System.out.println(x:"      |_____|      ");
13         System.out.println(x:"      |_____|      ");
14     }
15 }
16

PROBLEMS 10 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\kseilons\YandexDisk\university\3 кypc\Java\lesson> & 'C:\Program Files\Eclipse
ionMessages' '-cp' 'C:\Users\kseilons\AppData\Roaming\Code\User\workspaceStorage\6eff8cef751
problem.problem2'
      /\
      /  \
      /____\
      /|    |\
      /_|___|\
      |_____|
      |_____|
      |_____|
  
```

Рисунок 3.1– Дом нарисованный с помощью ASCII-графики

Проблема 3: The Snake Box Factory

Прочтите сценарий в обзоре и подумайте, какие объекты можно смоделировать в рамках создания программного решения. Определите 3 объекта в этом сценарии (помните, объекты могут быть материальными или абстрактными). Для каждого объекта перечислите 3 свойства и 3 варианта поведения.

Объект	Box	Order	Deliveryman
Свойства объекта	size	orderNumber	deliveryAreaId
	material	orderStatus	energy
	id	id	id
Варианты поведения	pack	create	move
	take	updateStatus	interactWithOtherSnakes
	unpack	cancel	deliver