Федеральное государственное автономное образовательное учреждение

высшего образования

Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Высшая школа «Киберфизические системы и управление»

**Отчет №7**

по дисциплине «Системный подход к разработке программного обеспечения»

|  |  |  |
| --- | --- | --- |
| Выполнил:  студент гр. 3530902/00201 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Позолотин О. В. |
|  | <*подпись*> |  |
|  |  |  |
|  |  |  |
| Руководитель:  Кандидат т.н. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Нестеров С. А. |
|  | <*подпись*> |  |

«\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2022 г.

Санкт-Петербург

2022

**Ход работы**

**По уроку 6.2**

**WhileLoopEx**

**SquareRootWhile**

**Изображение выглядит как текст

Автоматически созданное описание**

Рисунок 1 – код программы и результат ее работы

В ходе работы программы был использован цикл while для повторяющегося запроса числа от пользователя до тех пор, пока пользователь не введет положительное число.

**SumofNums**

**Изображение выглядит как текст

Автоматически созданное описание**

Рисунок 2 -код программы с использованием цикла while и результат работы

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 3 – код программы с использованием цикла do-while и результат работы

В данной программе цикл сделан двумя способами с предусловием и постусловием.

**По уроку 6.3**

**BreakContinueEx**

**ComputeSum**

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 4 – код программы и результат работы

Была написана программа, которая предлагает ввести пользователю десять чисел и посчитать их сумму. При этом если пользователь вводит ноль, программа сразу выводит сумму, не дожидаясь ввода дальнейших чисел, что реализовано за счет оператора break в теле цикла.

**CountChar**

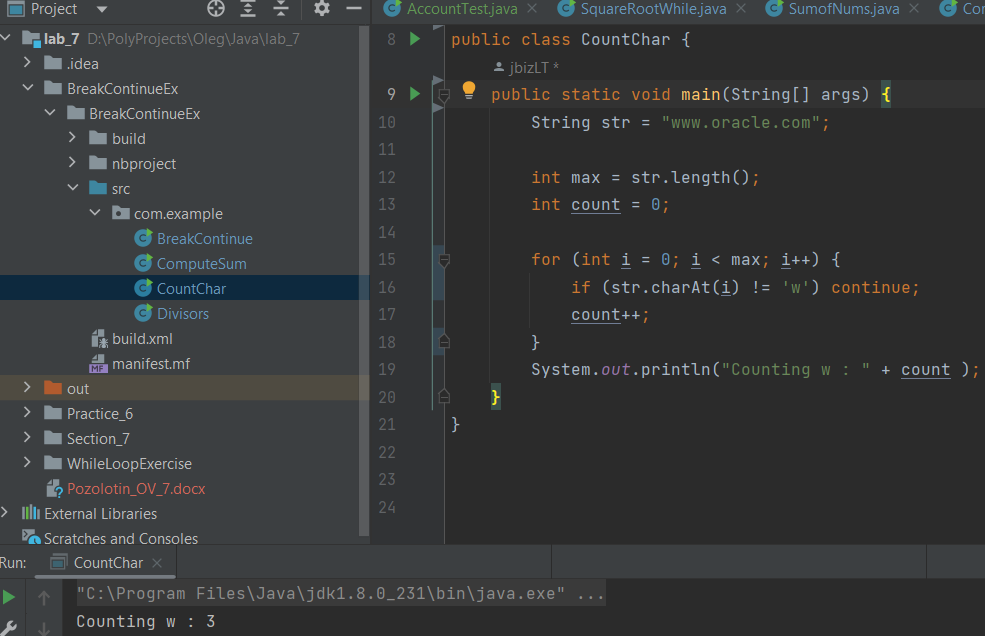
****

Рисунок 5 – код программы и результат работы

С помощью оператора continue была написана программа, реализующая подсчет символов “w” в строке. Оператор continue пропускает итерацию цикла, но, в отличие от оператора break, не завершает сам цикл.

**BreakContinue**

**Изображение выглядит как текст

Автоматически созданное описание**

Рисунок 6 – код программы и результат работы

Правильно расставленные операторы break и continue помогли реализовать требуемый функционал программы: выводить на экран только нечетные значения и останавливать цикл, когда значение счетчика равно 7.

**Divisors**

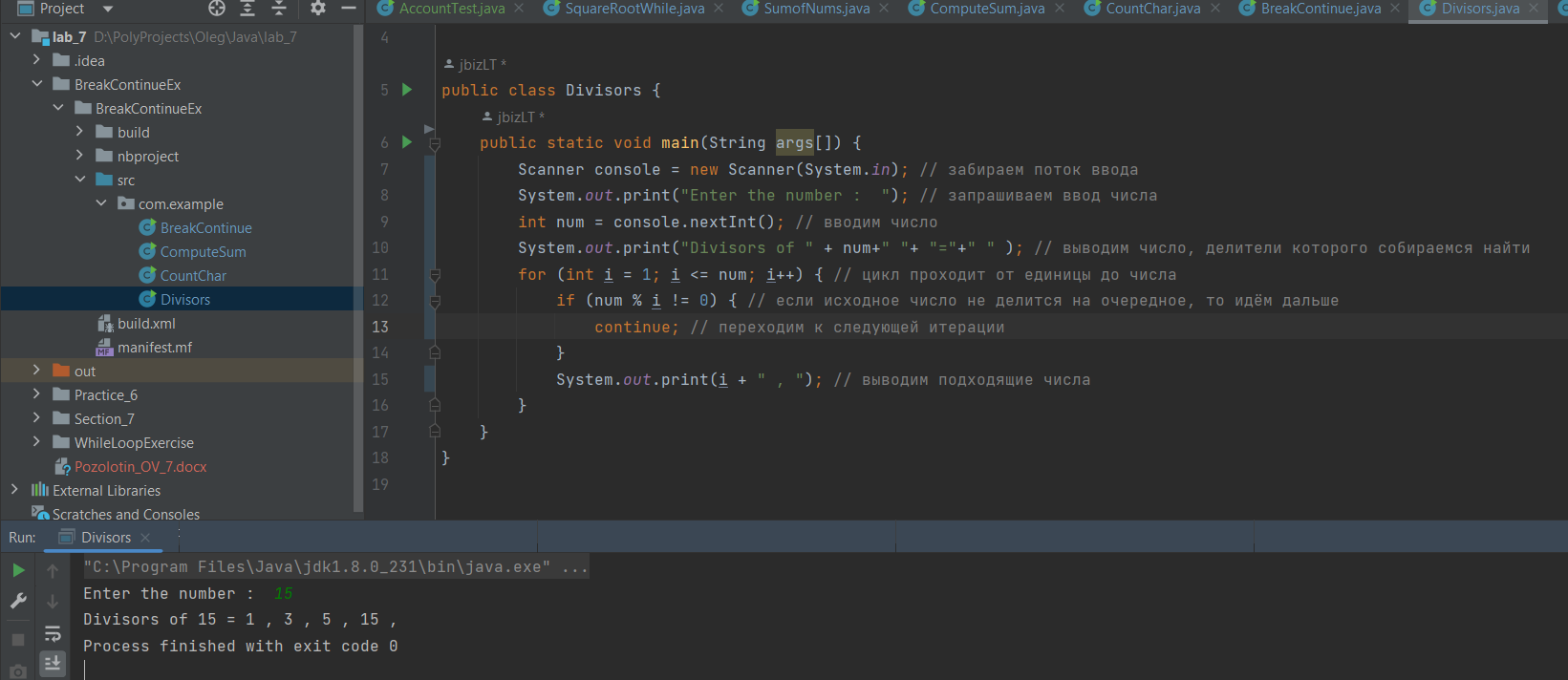


Рисунок 7 – программа для поиска делителей числа с комментариями

К программе добавлены комментарии к каждой строчке для пояснения производимых над ней действий.

**Практика 6**

**ProblemSet6\_1**

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 8 – код программы и результат работы

С помощью цикла while была написана программа, запрашивающая у пользователя пин-код и сравнивающая его с правильным. Если пользователь вводил неверный пин-код, программа уведомляла его об этом и предоставляла вторую попытку. Когда пользователь вводит верный пин-код, программа сообщает, что доступ к данным предоставлен.

ProblemSet6\_2

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 9 – код программы и результат работы

С помощью цикла for написана программа, находящая кратные числа, введенного пользователем.

**ProblemSet6\_3**

**Изображение выглядит как текст

Автоматически созданное описание**

Рисунок 10 – код метода createRectangle

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 11 – код метода createTriangle

Тестирование

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 12 – случай, когда стороны меньше 1

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 13 – случай, когда все стороны 1

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 14 – стандартный случай

**Занятие 7.1**

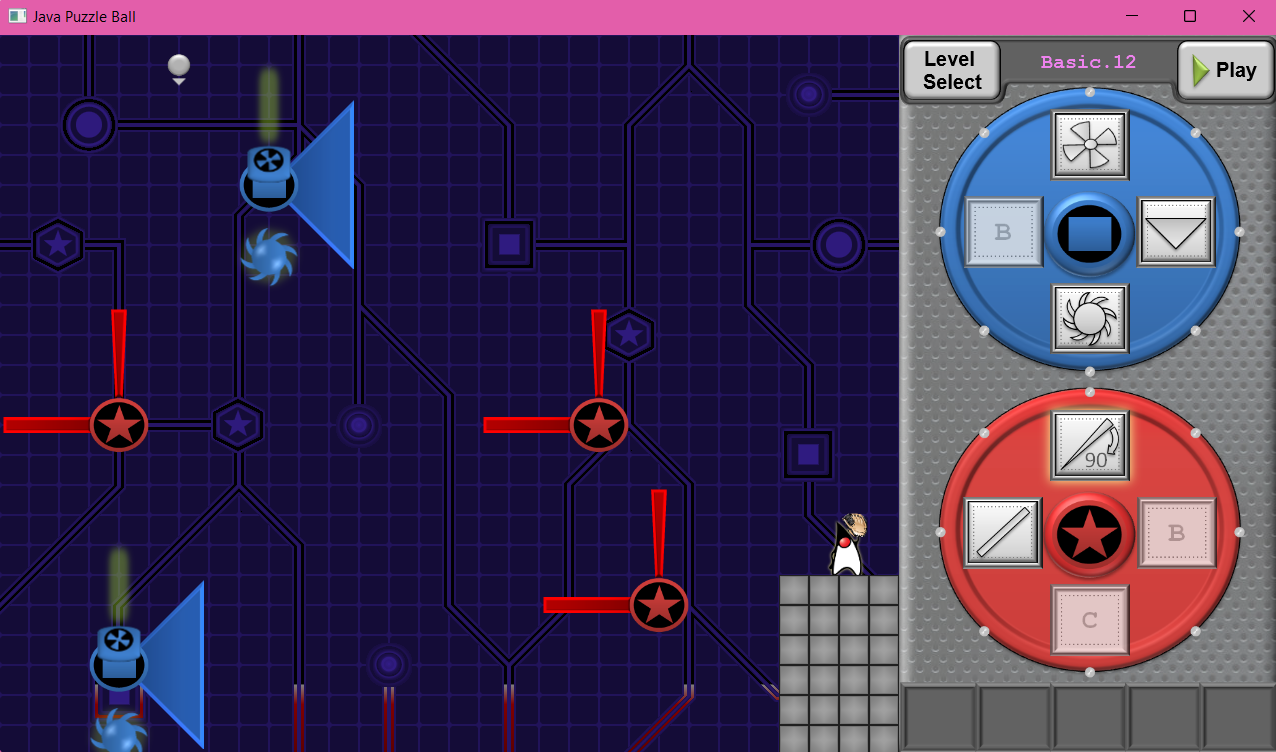
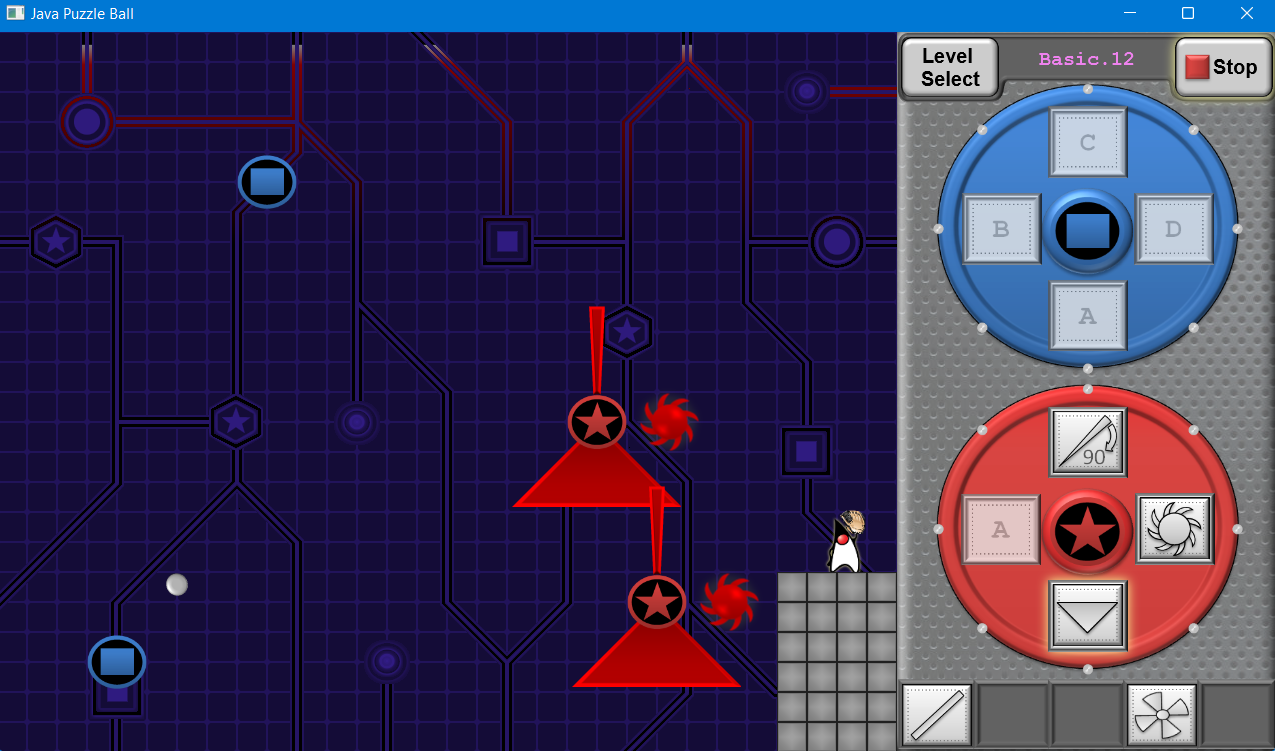


Рисунок 15–12 уровень в игре

На уровне представлено 3 объекта RedBumper, причем каждый экземпляр этих объектов имеет одинаковое поведение при взаимодействии с мячом. Примеры поведения: отбивания мяча от треугольной или простой стены.

У RedBumper есть поля и свойства. Поля включают в себя цвет, форму, положение, а свойства включают в себя воспроизведение звука удара,

вспышку, отбивание мяча или уничтожение. Например, объект RedBumper имеет метод для столкновений. В зависимости от того, является ли мяч мечом, объект либо отражает мяч, либо самоуничтожается под действием меча, как показано на рисунке 18. Код метода приведен на рисунке 19.



Изображение выглядит как текст

Автоматически созданное описаниеРисунок 16 – объект RedBumper уничтожился после столкновения с мечом

Рисунок 17 – код метода, отвечающего за столкновения с мячом

Рисунок 16 является подтверждением правильной работы метода, представленного на рисунке 18.

**CheckingAccount**

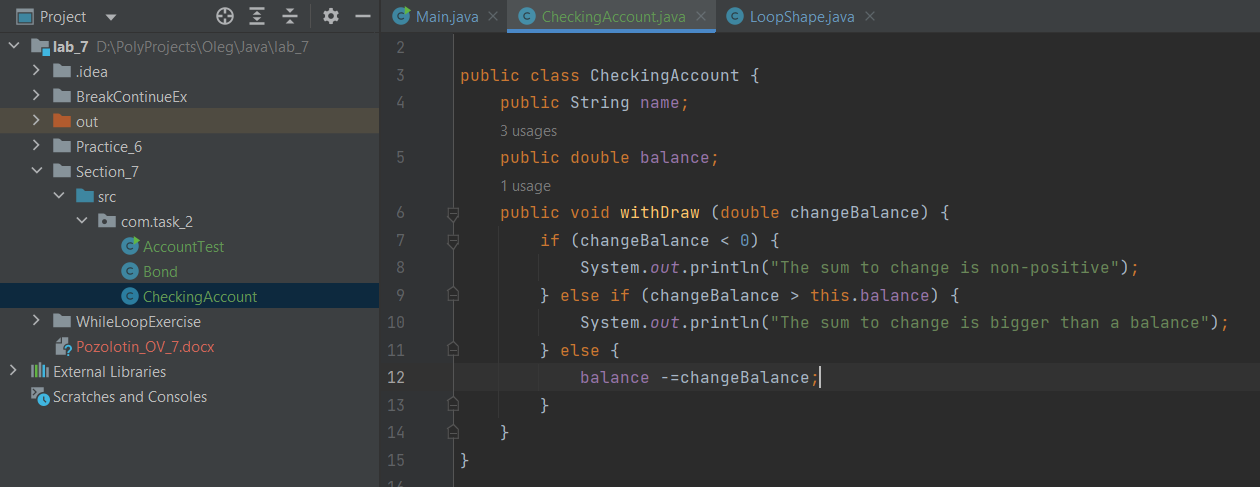
****

Рисунок 18 – код класса CheckingAccount

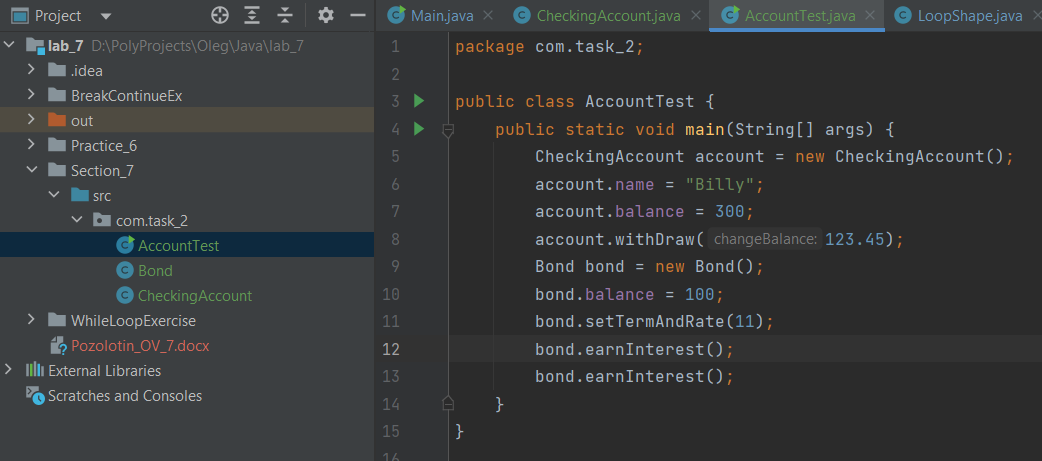


Рисунок 19 – код класса AccountTest

В ходе выполнения задания было создано 2 класса. Из главного метода 1 класса менялись значения полей 2 класса.

В папке проекта содержится папка src, содержащая папку package с названием accounttest, который, в свою очередь содержит файлы созданных двух классов.

**Упражнение 3**

**Изображение выглядит как текст

Автоматически созданное описание**

Рисунок 20 – код метода withdraw класса CheckingAccount

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 21 – 3 вызовы метода и результат работы

В ходе выполнения задания был написан метод для снятия средств со счета.

Код класса Bond:

public class Bond {

public String name;

public double balance, rate;

public int term, monthsRemaining;

public void setTermAndRate(int t){

if(t>=0 && t<12)

rate = 0.005;

else if(t>=12 && t<24)

rate = 0.010;

else if(t>=24 && t<36)

rate = 0.015;

else if(t>=36 && t<48)

rate = 0.020;

else if(t>=48 && t<=60)

rate = 0.025;

else{

System.out.println("Недопустимый срок");

t = 0;

}

term = t;

monthsRemaining = t;

}

public void earnInterest(){

if(monthsRemaining > 0){

balance += balance\*rate/12;

monthsRemaining--;

System.out.println("Баланс: $" +balance);

System.out.println("Ставка: " +rate);

System.out.println("Остаток месяцев:"

+monthsRemaining);

}

else{

System.out.println("Срок платежа облигации наступил");

}

}

}

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 22 – код программы и результат работы

Создан класс Bond – облигации.