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 №5**

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**

**4.4**

**RandomEx**

FlipCoin

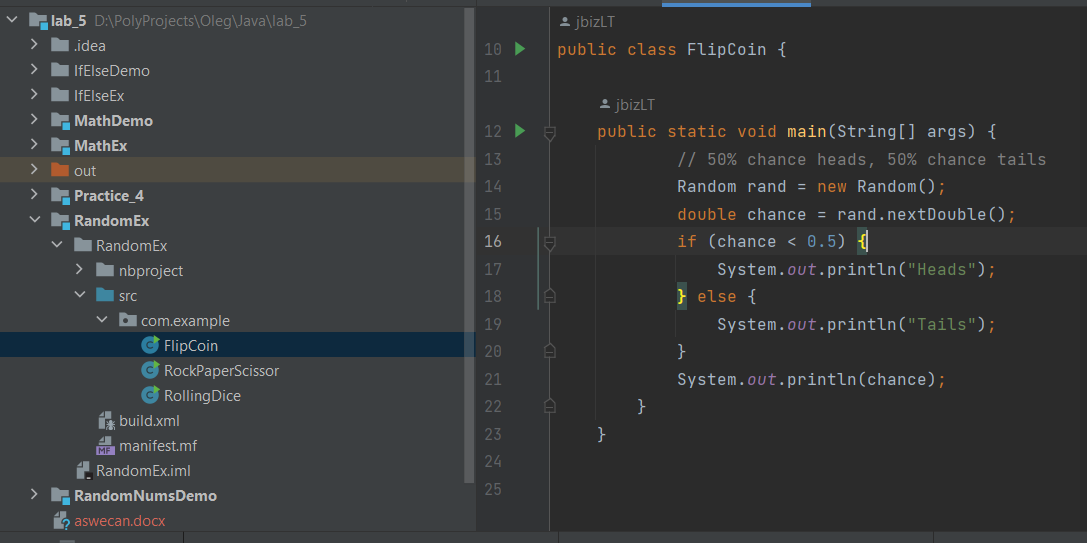


Рисунок 1 – код программы и возможный результат работы

В данном примере представлено использование класса Random и его метода nextDouble(), возвращающего случайное вещественное число в пределах от 0 до 1. Если вернулось число, меньшее 0,5, то выводится “heads”, “tails” – если большее 0,5.

**RockPaperScissor**

**Изображение выглядит как текст

Автоматически созданное описание**

Рисунок 2 – код программы и возможный результат работы

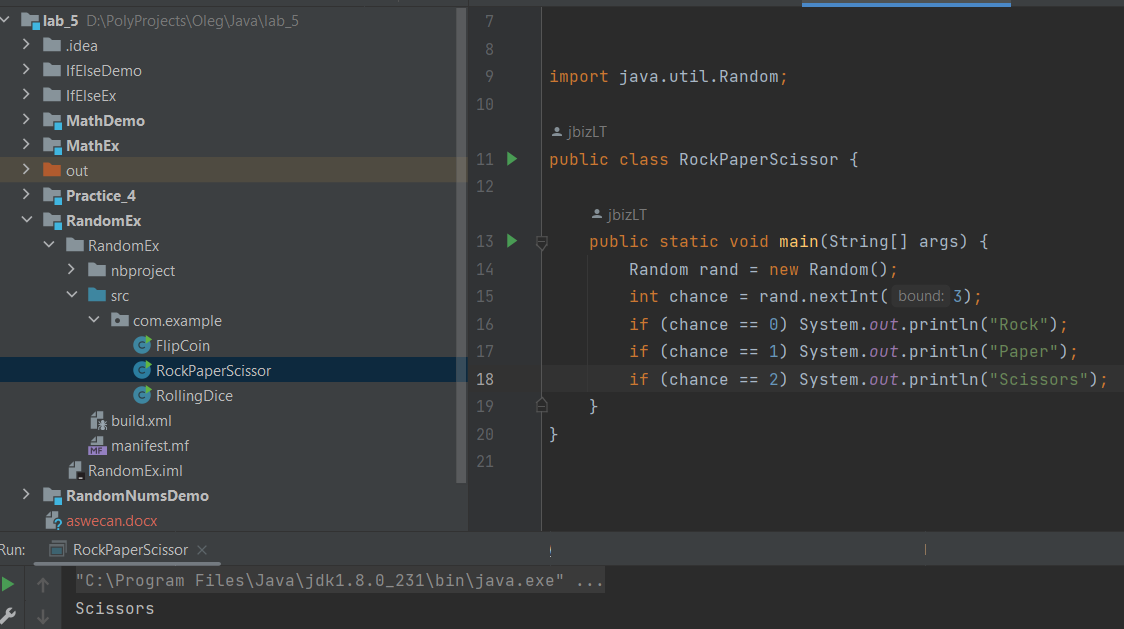


Рисунок 3 – еще один возможный вариант результата работы программы

В данном задании бралось случайное число из ограниченного диапазона (от 0 до 3). В зависимости от результата, выводились разные варианты хода в игре камень-ножницы-бумага.

**RandomNumsDemo**

**RockPaperScissor**

**Изображение выглядит как текст

Автоматически созданное описание**

Рисунок 4 – код программы и возможный результат работы

Программы выполняет ту же работы, что и написанная мной. Отличия – не выводит значение num и результат работы пишется на английском языке.

**RandomSeriesSeed**

**Изображение выглядит как текст

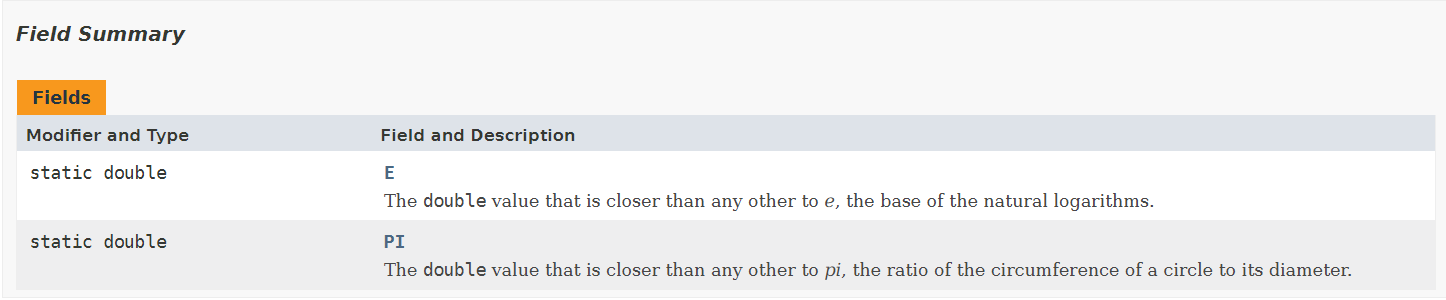
Автоматически созданное описание**

Рисунок 5 – код программы и возможный результат работы

В данном примере показано, как мы передаем параметру класса Random целочисленную константу в качестве ядра, а затем изменяем ее с помощью метода setSeed(). В результате работы программы видно, что при передаче одинаковых ядер генерируется одна и та же последовательность случайных чисел, при передаче другого ядра последовательность меняется.

**4.5**

**Документация класса Math**

****

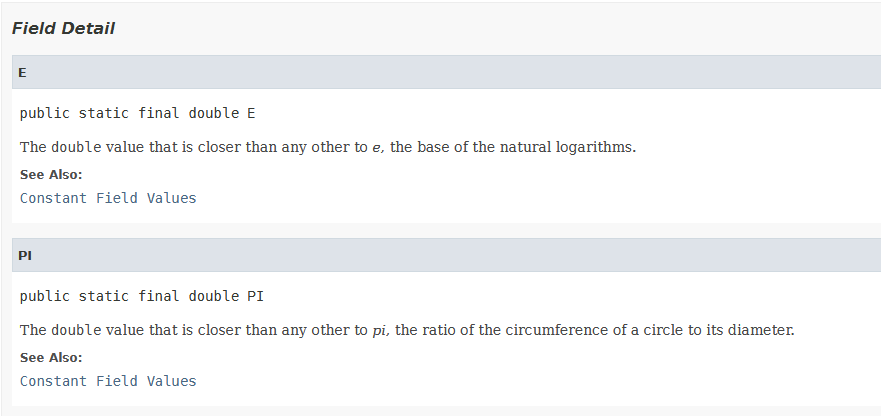
Рисунок 6 – поля класса Math

Рисунок 7 – описание полей класса Math

В полях класса Math объявлены 2 константные переменные: E и PI.

* Public static final double E – значение, которое ближе, чем любое другое, к e, основанию натуральных логарифмов.
* Public static final double PI - значение, которое ближе всего к пи, отношение длины окружности к ее диаметру.

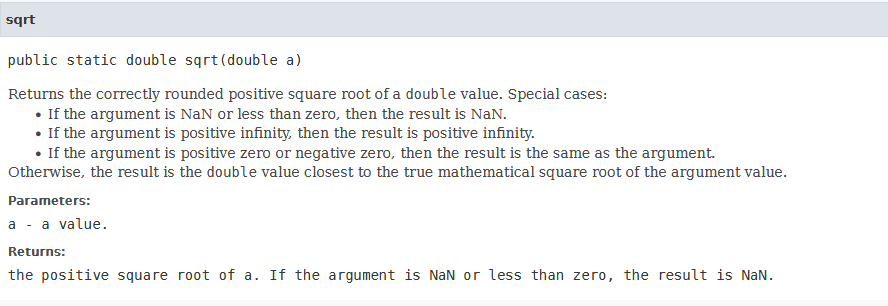


Рисунок 8 – документация по функции вычисления квадратного корня

Возвращаемое значение функции - положительный квадратный корень из а. Если аргумент равен NaN или меньше нуля, результатом будет NaN.

**MathDemo**

**PrintMathResult**

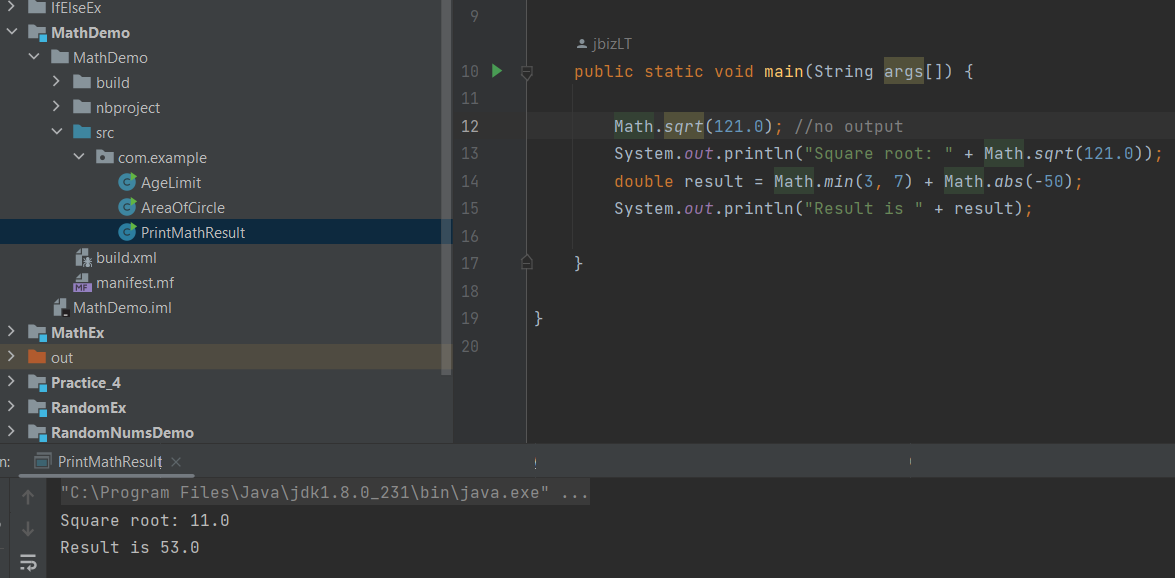
****

Рисунок 9 – код программы и результат его работы

В проекте использованы статические метода класса Math. Статические методы можно вызывать без создания экземпляра класса. В примере показано, что сами по себе статические методы результат никуда не выводят, но мы может сделать это сами с помощью System.out.println или с сохранением результата работы метода в переменную с последующим ее выводом с помощью System.out.println.

**AreaOfCircle**

**Изображение выглядит как текст

Автоматически созданное описание**

Рисунок 10 – код программы и результат ее работы

В программе запрашивается число – радиус окружности. Далее программа считает площадь круга, используя статическую переменную PI и выводит результат на экран.

**Упражнение 2**

* Math.abs(-1.23) = 1.23
* Math.pow(3, 2) = 32=9
* Math.sqrt(121.0) - Math.sqrt(256.0) = 11-16 = -5
* Math.abs(Math.min(-3, -5)) = |-5| = 5

**Упражнение 3**

* – Какое выражение позволит заменить отрицательные
* значения на 0 – Math.max(age, 0);
* – Какое выражение позволит округлить все большие
* значения до 40? – Math.min(40, age);

**AgeLimit**

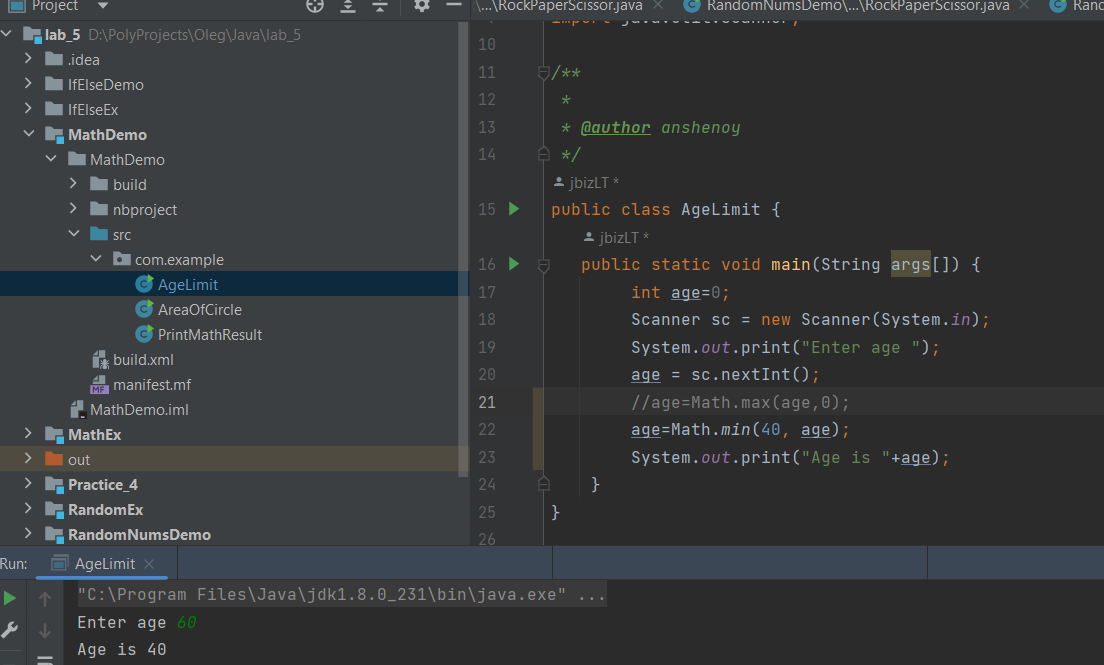


Рисунок 11 – код программы и результат ее работы

Написанные вручную выражения совпали с представленными в программе и правильно выполняют свою работу.

**MathEx**

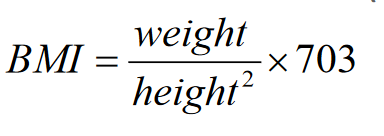
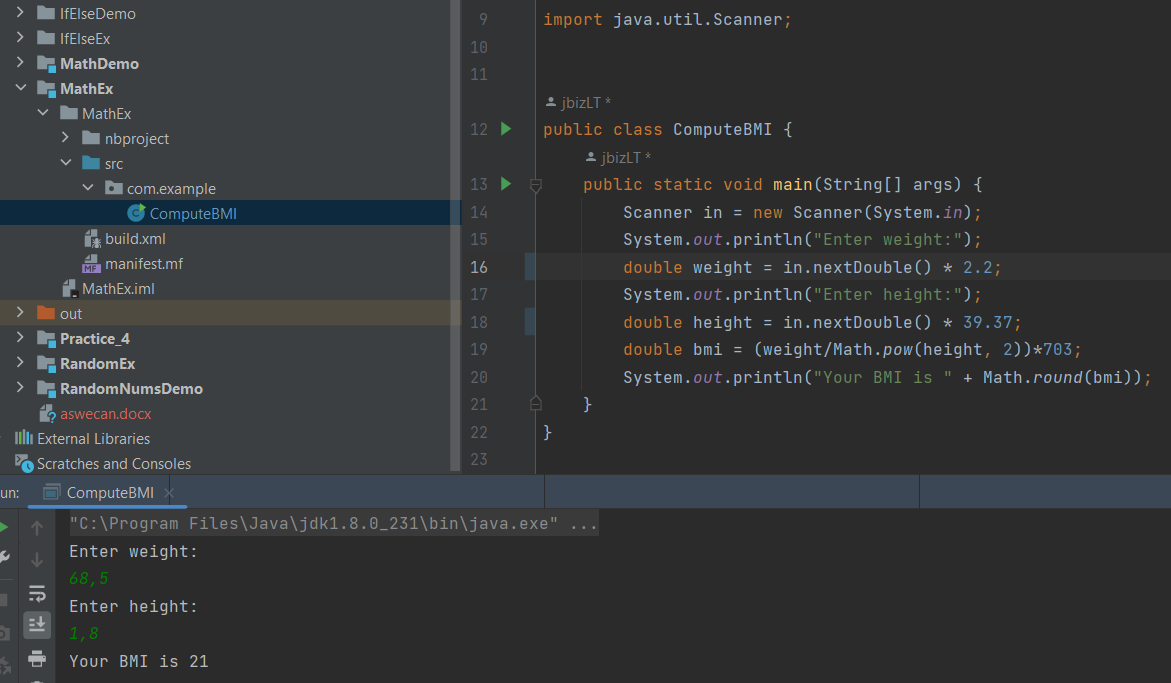
****

Рисунок 12 – формула для вычисления индекса массы человека

Рисунок 12 – код программы, ввод и результат

В ходе задания я, используя метод pow класса Math, рассчитывал индекс массы тела исходя от введенных пользователем данных.

**Практика 4**

Ниже рассмотрены менее тривиальные примеры из представленных.

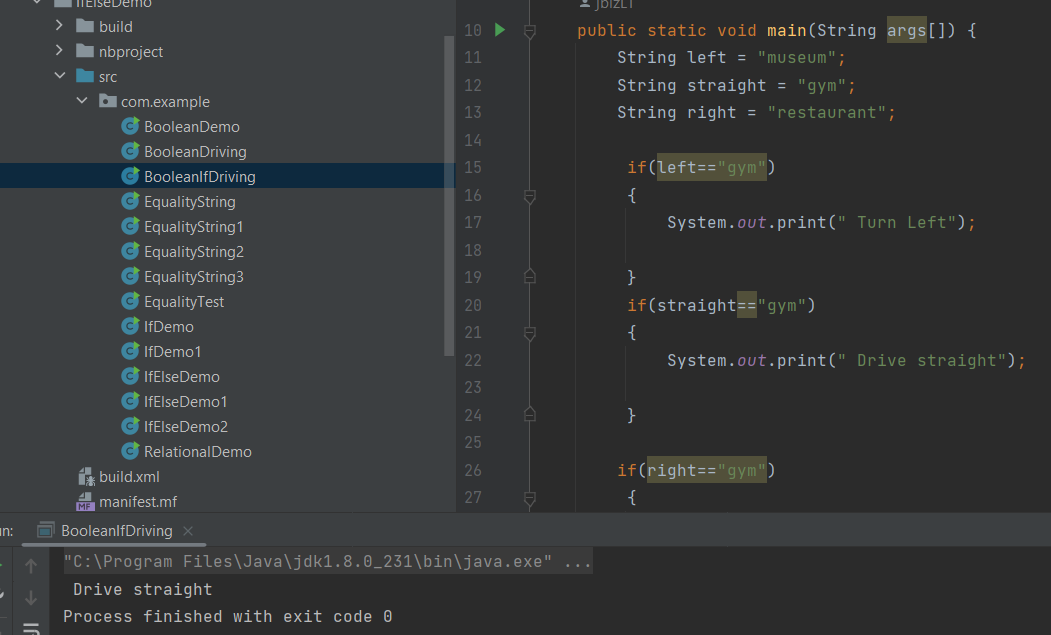


Рисунок 13 – код программы BooleanIfDriving и результат ее работы

Вывод программы зависит от значения переменных left, straight, right.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 14 – код программы EqualityString1

Программа использует конструкцию if-else для вывода результата, равны строки (без учета регистра) или нет. Также в программе присутствует метод класса String – equalsIgnoreCase.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 15 – код программы EqualityTest и результат ее работы

В программе реализована инициализация переменных типа Boolean логическими выражениями.

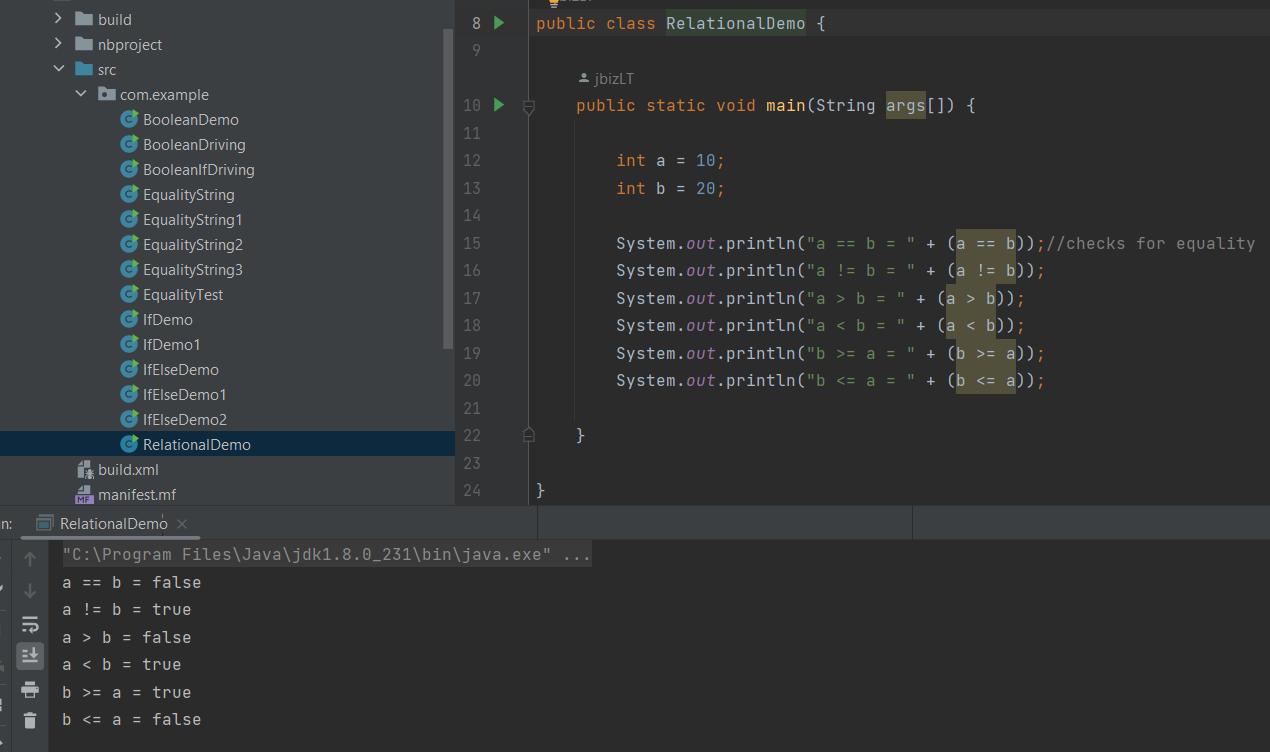


Рисунок 16 – код программы RelationalDemo и результат ее работы

В программе представлены все виды операторов отношения.

**IfElseEx**

**AgeValidity**

**Изображение выглядит как текст

Автоматически созданное описание**

Рисунок 17 – код программы и результат ее работы

В программе ожидался ввод от пользователя, а затем в переменную drivingUnderAge записывался результат сравнения введенного числа с 18. Если число меньше, значение становилось true, если нет – false.

**ChkOddEven**

**Изображение выглядит как текст

Автоматически созданное описание**

Рисунок 17 – код программы и результат ее работы

На вход от пользователя подавалось число. Программа, используя конструкцию if-else, делала вывод о том, четное число или нет.

**AgeCheck**

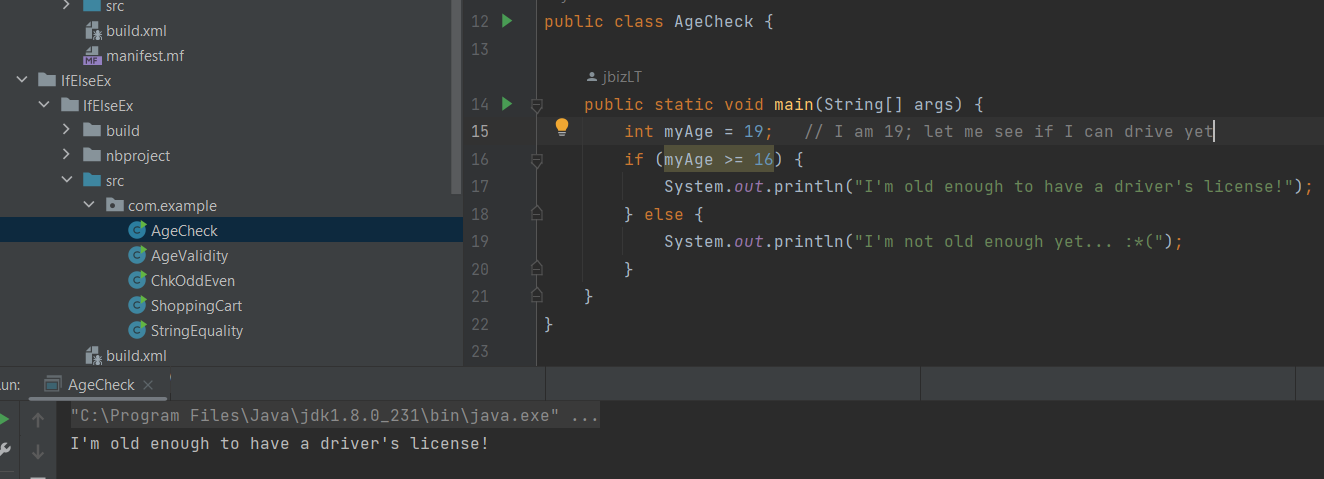
****

Рисунок 18 – код программы и результат ее работы

В программе была логическая ошибка. Если пользователю 16 лет, срабатывали сразу два оператора. После исправления такой ошибки нет, программа при всех вводимых значениях отрабатывает правильно.

**ShoppingCart**

**Изображение выглядит как текст

Автоматически созданное описание**

Рисунок 19 – дописанная часть программы и результат работы программы

В программе была использована конструкция if-else if – else для корректного отображения сообщения, зависящего от количества товара.

**StringEquality**

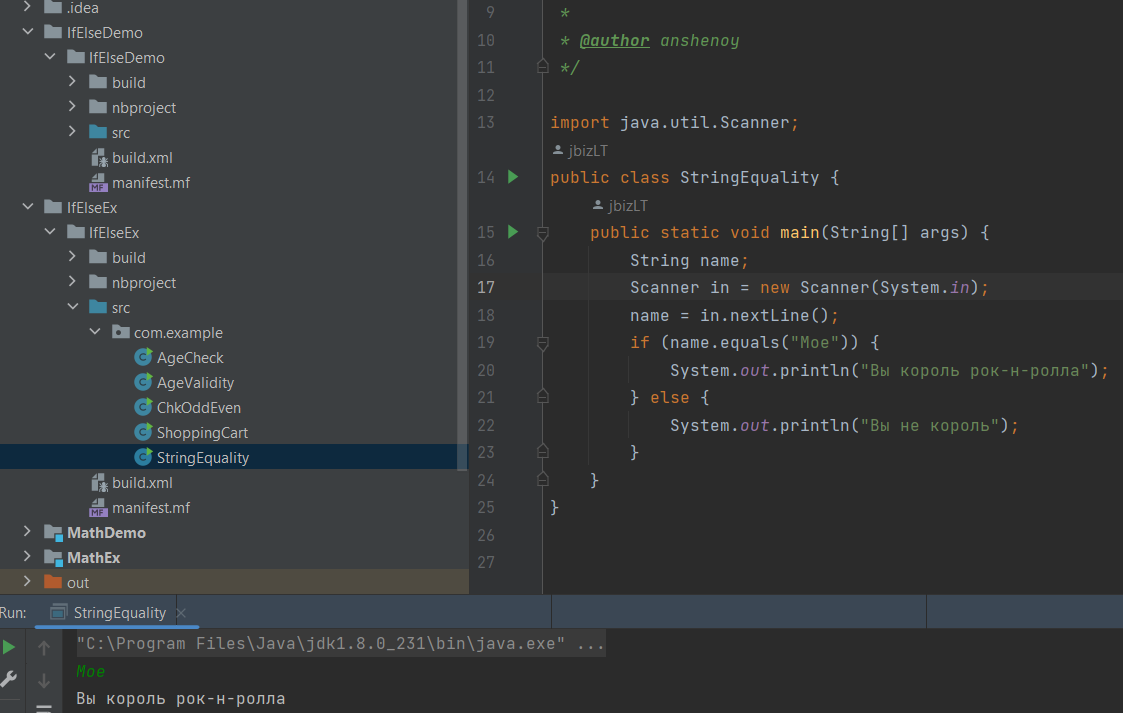
****

Рисунок 19 – код программы и результат его работы

В данном задании была работы со сравнением строк. Строки нельзя сравнивать через обычные операторы, например, = = из-за того, что String – объект, а не примитив, и из-за этого при попытке сравнения строк на равенство с помощью оператора = =, будет сравниваться не содержимое строк, а адрес.

**Практика 4**

**Изображение выглядит как текст

Автоматически созданное описание**

Рисунок 20 – код ComputeMethods

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 21 – код TestClass и результат работы программы

В ходе выполнения программы мной были написаны несколько методов, реализующие разный функционал с использованием методов класса Math, Random.

В ходе выполнения этого задания я использовал методы класса String :

Split() и CharAt().

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 23 – запись на курс по UML