

Решение задач из лекций 10 и 11

В случае если вы претерпели сложности при их решении

Задача 1 из 10-ой лекции

```
1  public class Main {
2
3  public static void main(String[] args) {
4      chooseToDrink( isTodayFriday: true,  aLotOfMoney: true);
5      chooseToDrink( isTodayFriday: true,  aLotOfMoney: false);
6      chooseToDrink( isTodayFriday: false, aLotOfMoney: true);
7      chooseToDrink( isTodayFriday: false, aLotOfMoney: false);
8  }
9
10 private static void chooseToDrink(boolean isTodayFriday, boolean aLotOfMoney) {
11     if (aLotOfMoney) {
12         print("I am going to drink");
13     } else if (isTodayFriday) {
14         print("Gonna lend some money and go to drink");
15     } else {
16         print("Just go home and eat some food");
17     }
18 }
19
20 private static void print(String text) {
21     System.out.println(text);
22 }
23 }
```

Main > chooseToDrink()

Run: Main x

```
/usr/lib/jvm/java-1.8.0-openjdk-amd64/bin/java ...
I am going to drink
Gonna lend some money and go to drink
I am going to drink
Just go home and eat some food
Process finished with exit code 0
```

Получается, что мы идем пить пиво если денег много или же если пятница, значит сначала проверим есть ли у нас деньги вне зависимости от дня недели. Иначе, если денег нет, но сегодня пятница, то мы идем одалживать и пить. И все остальные случаи это когда нет денег и на дворе не пятница. В мейне вызываем метод со всеми возможными комбинациями 2 булеанов и видим что все написали правильно.

Задача 2 из 10-ой лекции

Сначала мы напишем очень просто – с помощью if else без && и вложенности.

Все конечно же работает, но можно сделать иначе.

Сначала проверим что денег хватает хоть на что-то (меньшее из 2), тогда мы точно что-то выберем, иначе сразу домой. После уже проверим сколько у нас денег , хватает на виски и если остатка хватит еще и на пиво, то его тоже возьмем. Здесь более логично, согласны?

```

1 ▶ public class Main {
2
3     private static final int BEER_COST = 5;
4     private static final int WHISKEY_COST = 9;
5
6 ▶     public static void main(String[] args) {
7         chooseToDrink(BEER_COST, WHISKEY_COST, money: 15);
8         chooseToDrink(BEER_COST, WHISKEY_COST, money: 10);
9         chooseToDrink(BEER_COST, WHISKEY_COST, money: 7);
10        chooseToDrink(BEER_COST, WHISKEY_COST, money: 2);
11    }
12
13    private static void chooseToDrink(int beerCost, int whiskeyCost, int money) {
14        if (money >= beerCost + whiskeyCost) {
15            print("whiskey and beer, please!");
16        } else if (money > whiskeyCost) {
17            print("whiskey, please!");
18        } else if (money > beerCost) {
19            print("beer, please!");
20        } else {
21            print("oops, seems like I have no enough money today");
22        }
23    }
24
25    private static void print(String text) {
26        System.out.println(text);
27    }
28 }
29

```

Run: Main ×

```

/usr/lib/jvm/java-1.8.0-openjdk-amd64/bin/java ...
whiskey and beer, please!
whiskey, please!
beer, please!
oops, seems like I have no enough money today
Process finished with exit code 0

```

```

private static void chooseToDrink(int beerCost, int whiskeyCost, int money) {
    if (money >= beerCost) {
        if (money >= whiskeyCost) {
            print("whiskey, please!");
            int rest = money - whiskeyCost;
            if (rest >= beerCost) {
                print("beer, please!");
            }
        } else {
            print("beer, please!");
        }
    } else {
        print("oops, seems like I have no enough money today");
    }
}

```

Задача 1 из 11-ой лекции

```
1 public class Main {
2
3     public static void main(String[] args) {
4         print(triangleExists( a: 3, b: 4, c: 5));
5         print(triangleExists( a: 1, b: 2, c: 3));
6     }
7
8     private static boolean triangleExists(int a, int b, int c) {
9         return a + b > c && b + c > a && a + c > b;
10    }
11
12    private static void print(boolean b) {
13        System.out.println(b);
14    }
15 }
```

Main > main()

Run: Main x

/usr/lib/jvm/java-1.8.0-openjdk-amd64/bin/java ...
true
false
Process finished with exit code 0

На самом деле нам нужна еще проверка, что длина сторон неотрицательна, ведь можно передать в аргумент как ноль так и отрицательные числа. Так что ваш метод должен быть лучше.

```
1 public class Main {
2
3     public static void main(String[] args) {
4         print(triangleExists( a: 3, b: 4, c: 5));
5         print(triangleExists( a: 3, b: -4, c: 5));
6         print(triangleExists( a: 0, b: 2, c: 3));
7     }
8
9     private static boolean triangleExists(int a, int b, int c) {
10        return a > 0 && b > 0 && c > 0 &&
11               a + b > c && b + c > a && a + c > b;
12    }
13 }
```

Main > main()

Run: Main x

/usr/lib/jvm/java-1.8.0-openjdk-amd64/bin/java ...
true
false
false
Process finished with exit code 0

Задача 2 из 11-ой лекции

```
1 public class Main {
2
3     public static void main(String[] args) {
4         print(min(a: 1, b: 2, c: 3, d: 4));
5         print(min(a: -5, b: -20, c: 0, d: 9));
6     }
7
8     private static int min(int a, int b, int c, int d) {
9         int minAB = min(a, b);
10        int minCD = min(c, d);
11        return min(minAB, minCD);
12    }
13
14    private static int min(int a, int b) {
15        return a < b ? a : b;
16    }
17
18    private static void print(Object b) {
19        System.out.println(b);
20    }
21 }
```

Main

Run: Main x

/usr/lib/jvm/java-1.8.0-openjdk-amd64/bin/java ...

1

-20

Process finished with exit code 0

Алгоритм у нас простой. Написали метод нахождения минимума из 2 чисел и используем для нахождения из 4. Итого сравниваем меньшее из первой пары и второй. Но давайте попробуем написать иначе, без использования другой функции.

Алгоритм теперь таков: предположим что наименьшее число из четырех это первое. Далее сравним с ним второе число. Если оно окажется меньше, то перепишем значение минимума. И так до конца. Вы можете написать этот метод совсем иначе. Нет одного верного варианта.

```
1 public class Main {
2
3     public static void main(String[] args) {
4         print(min(a: 1, b: 2, c: 3, d: 4));
5         print(min(a: -5, b: -20, c: 0, d: 9));
6     }
7
8     private static int min(int a, int b, int c, int d) {
9         int min = a;
10        if (b < min) {
11            min = b;
12        }
13        if (c < min) {
14            min = c;
15        }
16        if (d < min) {
17            min = d;
18        }
19        return min;
20    }
21}
```

Main

Run: Main x

/usr/lib/jvm/java-1.8.0-openjdk-amd64/bin/java ...

1

-20

Process finished with exit code 0

```
1 public class Main {
2
3     public static void main(String[] args) {
4         goToWork(isWeekend: true, isHoliday: true);
5         goToWork(isWeekend: true, isHoliday: false);
6         goToWork(isWeekend: false, isHoliday: true);
7         goToWork(isWeekend: false, isHoliday: false);
8     }
9
10    private static void goToWork(boolean isWeekend, boolean isHoliday) {
11        if (!isHoliday) {
12            if (isWeekend) {
13                print("Alarm on 11.00");
14            } else {
15                print("Alarm on 6.00! Go to work");
16            }
17        }
18    }
19}
```

Main

Run: Main x

/usr/lib/jvm/java-1.8.0-openjdk-amd64/bin/java ...

Alarm on 11.00

Alarm on 6.00! Go to work

Process finished with exit code 0

Задача 3 из 11-ой лекции

Сначала проверим что у нас не праздничный день. Далее если это выходной, то на 11 будильник. Иначе – рабочий день.

Задача 4 из 11-ой лекции

```
3  ▶ public static void main(String[] args) {
4      checkTriangleType( angleA: 100, angleB: 60, angleC: 20);
5      checkTriangleType( angleA: 90, angleB: 45, angleC: 45);
6      checkTriangleType( angleA: 60, angleB: 60, angleC: 60);
7      checkTriangleType( angleA: 20, angleB: 80, angleC: 80);
8      checkTriangleType( angleA: 70, angleB: 50, angleC: 60);
9      checkTriangleType( angleA: 100, angleB: 100, angleC: 20);
10     checkTriangleType( angleA: -1, angleB: 200, angleC: 0);
11     checkTriangleType( angleA: 60, angleB: 61, angleC: 20);
12 }
13
14 private static void checkTriangleType(int angleA, int angleB, int angleC) {
15     String type;
16     if (angleA > 0 && angleA < 180 &&
17         angleB > 0 && angleB < 180 &&
18         angleC > 0 && angleC < 180 &&
19         angleA + angleB + angleC == 180) {
20
21         if (angleA > 90 || angleB > 90 || angleC > 90) {
22             type = "тупоугольный";
23         } else if (angleA == angleB && angleB == angleC) {
24             type = "равносторонний";
25         } else if (angleA == 90 || angleB == 90 || angleC == 90) {
26             type = "прямоугольный";
27         } else if (angleA == angleB || angleA == angleC || angleB == angleC) {
28             type = "равнобедренный";
29         } else {
30             type = "остроугольный";
31         }
32     } else {
33         type = "невалидные данные";
34     }
35     print(type);
36 }
37
```

Main > checkTriangleType()

Run: Main x

/usr/lib/jvm/java-1.8.0-openjdk-amd64/bin/java ...

- тупоугольный
- прямоугольный
- равносторонний
- равнобедренный
- остроугольный
- невалидные данные
- невалидные данные
- невалидные данные

Для начала нужно проверить входные данные. Никогда не забывайте этого делать. Не ожидайте, что кто-то введет правильные числа. Во-первых человек может ошибиться просто введя 2 раза по 90, или же на вход придет 0. Поэтому пишем проверку. Все углы в треугольнике должны быть больше 0 и меньше 180. И конечно же сумма должна равняться 180. После уже просто проверим все углы на их значения. Заметьте, что данный код имеет приоритеты. Т.е. если у вас будет прямоугольный равнобедренный треугольник, то метод скажет только о том, что он прямоугольный. Как это исправить? Легко. Можно добавлять

значения по мере прохождения в этой цепи логики. Все же помнят как конкатенировать строки?

```
3 public static void main(String[] args) {
4     checkTriangleType( angleA: 100, angleB: 60, angleC: 20);
5     checkTriangleType( angleA: 90, angleB: 45, angleC: 45);
6     checkTriangleType( angleA: 60, angleB: 60, angleC: 60);
7     checkTriangleType( angleA: 20, angleB: 80, angleC: 80);
8     checkTriangleType( angleA: 70, angleB: 50, angleC: 60);
9     checkTriangleType( angleA: 100, angleB: 100, angleC: 20);
10    checkTriangleType( angleA: -1, angleB: 200, angleC: 0);
11    checkTriangleType( angleA: 60, angleB: 61, angleC: 20);
12
13
14    private static void checkTriangleType(int angleA, int angleB, int angleC) {
15        String type = "";
16        if (angleA > 0 && angleA < 180 &&
17            angleB > 0 && angleB < 180 &&
18            angleC > 0 && angleC < 180 &&
19            angleA + angleB + angleC == 180) {
20
21            if (angleA > 90 || angleB > 90 || angleC > 90) {
22                type += "тупоугольный ";
23            }
24            if (angleA == angleB && angleB == angleC) {
25                type += "равносторонний ";
26            }
27            if (angleA == 90 || angleB == 90 || angleC == 90) {
28                type += "прямоугольный ";
29            }
30            if (angleA == angleB || angleA == angleC || angleB == angleC) {
31                type += "равнобедренный ";
32            }
33            if (angleA < 90 && angleB < 90 && angleC < 90) {
34                type += "остроугольный ";
35            }
36        } else {
37            type = "невалидные данные";
38        }
39        print(type);
40    }
41}
```

Main > checkTriangleType()

Run: Main x

/usr/lib/jvm/java-1.8.0-openjdk-amd64/bin/java ...

тупоугольный
прямоугольный равнобедренный
равносторонний равнобедренный остроугольный
равнобедренный остроугольный
остроугольный
невалидные данные
невалидные данные
невалидные данные

Compilation completed successfully in 611 ms (moments ago) 34:40

Но заметьте какой я написал оператор += это означает сразу 2 действия, добавить к существующей строке новую и присвоить существующее. т.е. `text += "a";` то же самое что и `text = text + "a";`

Задача 5 из 11-ой лекции

Мне кажется если человеку больше 100 лет, то он вряд-ли пользуется интернетом и нашими программами. Но вы сами решаете какой возраст валидный.

```
3  ▶ public static void main(String[] args) {
4      checkAge(18);
5      checkAge(16);
6      checkAge(-2);
7      checkAge(532);
8  }
9
10 private static void checkAge(int age) {
11     if (age < 0 || age > 100) {
12         print("invalid age");
13     } else {
14         if (age >= 18) {
15             print("Access is granted");
16         } else {
17             print("Access is denied");
18         }
19     }
20 }
21
```

Main > main()

Run: Main ×

/usr/lib/jvm/java-1.8.0-openjdk-amd64/bin/jav
Access is granted
Access is denied
invalid age
invalid age
Process finished with exit code 0

Задача 6 из 11-ой лекции

В этой задаче проще всего сначала также отсеять невалидные данные, и после уже приступить к основной логике. Вообще старайтесь первым же делом проверить входные данные (аргументы) на валидность и уже потом переходите к сути.

В этом случае проще всего идти от наименьшего возраста. Так как мы сначала определяем минимальный возраст, то все остальные if простые. Можете попробовать сначала обработать случай с пенсионером и потом уже идти вниз по возрасту. Вы увидите какой некрасивый код у вас получается.


```
2
3 public static void main(String[] args) {
4     checkAge(5);
5     checkAge(12);
6     checkAge(16);
7     checkAge(18);
8     checkAge(21);
9     checkAge(30);
10    checkAge(70);
11    checkAge(1010);
12    checkAge(-100);
13 }
14
15 private static void checkAge(int age) {
16     if (age < 0 || age > 100) {
17         print("невалидный возраст");
18     } else {
19         if (age < 6) {
20             print("в детсад");
21         } else if (age < 15) {
22             print("в школу");
23         } else if (age < 17) {
24             print("в колледж");
25         } else if (age < 22) {
26             print("в вуз");
27         } else if (age < 65) {
28             print("на работу");
29         } else {
30             print("в поликлинику");
31         }
32     }
33 }
```

Main > main()

Run: Main x

/usr/lib/jvm/java-1.8.0-openjdk-amd64/bin/ja

в детсад
в школу
в колледж
в вуз
в вуз
на работу
в поликлинику
невалидный возраст
невалидный возраст

Задача 7 из 11-ой лекции

В этой задаче мы просто сравним аргумент с константами. Мы говорили, что сравнивать через оператор == можно все что угодно. Но как видим среда разработки выдала нам подсказку – если поставить курсор на == String values are compared using '==' not equals. Что за equals? Скорей всего это метод который сравнивает строки. Нажимаем Alt+Enter на операторе и получаем замену через equals. По сути ничего не поменялось. И я вам расскажу

про методы класса String в следующих лекциях. Постарайтесь не забыть эту задачу, мы к ней вернемся в други лекциях.

```
1 public class Main {
2
3     private static final String WELCOME = "Добро пожаловать";
4
5     public static void main(String[] args) {
6         checkDenisOrAnton( name: "Антон");
7         checkDenisOrAnton( name: "Кирилл");
8     }
9
10    private static void checkDenisOrAnton(String name) {
11        if (name == "Денис" || name == "Антон") {
12            print(WELCOME + " отсюда");
13        } else {
14            print(WELCOME);
15        }
16    }
17 }
```

Main > checkDenisOrAnton()

Run: Main x

/usr/lib/jvm/java-1.8.0-openjdk-amd64/bin/java ...

Добро пожаловать отсюда

Добро пожаловать

Process finished with exit code 0

Задача 8 из 11-ой лекции

Итак, мы создаем переменную над методом мейн и следовательно можем ее использовать в других методах. В нашем методе мы проверяем, что оно меньше 5 и увеличиваем на 1. Т.е. код на линии 17 означает `count = count + 1`; т.е. мы берем значение которое было до этой линии, добавляем к нему 1 и присваиваем в переменную. Можете пройти дебагером и посмотреть значение переменной после каждого вызова. Сначала значение 0, после мы вызываем метод и в конце уже значение 1. В следующий раз, т.е. второй раз, у нас значение меняется с 1 на 2. И так мы доходим до случая, когда на 6-ой раз у нас переменная равна 5. А как мы знаем $5 < 5$ это false. И кстати да, мы считаем в программировании с нуля. Это очень удобно. Хочешь 5 вызовов функции – пиши `count < 5` и инициализируй ее `count = 0`;

Подробнее мы вернемся к этой задаче в других лекциях и разберем ее модификацию. Как сразу вызвать метод 5 раз.

```
1 public class Main {
2
3     private static int count = 0;
4
5     public static void main(String[] args) {
6         increment();
7         increment();
8         increment();
9         increment();
10        increment();
11        increment();
12    }
13
14    private static void increment() {
15        if (count < 5) {
16            print("вызов метода");
17            count++;
18        } else {
19            print("вы превысили лимит");
20        }
21    }
22
23    private static void print(Object b) {
24        System.out.println(b);
25    }
26 }
```

Main > main()

Run: Main x

/usr/lib/jvm/java-1.8.0-openjdk-amd64/bin/j

вызов метода
вызов метода
вызов метода
вызов метода
вызов метода
вы превысили лимит

Process finished with exit code 0

Задача 9 из 11-ой лекции

Ну здесь совсем просто. Если температура ниже нуля то лед, до 100 это вода и все что выше пар. Вы можете для разнообразия попробовать написать иначе. Например так

if (temp > 100)

else if (temp > 0)

else

Получится то же самое.

```
3 public static void main(String[] args) {
4     waterState( temp: -1);
5     waterState( temp: 10);
6     waterState( temp: 120);
7 }
8
9 private static void waterState(int temp) {
10     if (temp < 0) {
11         print("ice");
12     } else if (temp < 100) {
13         print("liquid");
14     } else {
15         print("vapor");
16     }
17 }
18
```

Main > waterState()

Run: Main x

/usr/lib/jvm/java-1.8.0-openjdk-amd64/bin/java

ice

liquid

vapor

Process finished with exit code 0

Задача 10 из 11-ой лекции

```
3 public static void main(String[] args) {
4     noLGBT( isGender1Male: true, isGender2Male: true);
5     noLGBT( isGender1Male: true, isGender2Male: false);
6     noLGBT( isGender1Male: false, isGender2Male: true);
7     noLGBT( isGender1Male: false, isGender2Male: false);
8 }
9
10 private static void noLGBT(boolean isGender1Male, boolean isGender2Male) {
11     String result = isGender1Male != isGender2Male ? "Совет да любовь" : "Валите в США!";
12     print(result);
13 }
14
```

Main > main()

Run: Main x

/usr/lib/jvm/java-1.8.0-openjdk-amd64/bin/java ...

Валите в США!

Совет да любовь

Совет да любовь

Валите в США!

Здесь мы принимаем 2 параметра – пол первого человека и пол второго. Я написал через тернарный оператор одним условием и попробую объяснить как это работает. $A \neq B$ значит для boolean типов что они одновременно не true или false, т.е. $true \neq false$ или $false \neq true$.

Вы бы могли написать иначе, в несколько строк. Но это считается самым эффективным способом. Как видите все работает. Пары МЖ и ЖМ прекрасно отрабатывают. А вот пары ММ и ЖЖ не проходят проверку.

п.с. согласен, задача может кого-то оскорбить, но это же просто кодинг. И на таких простых примерах можно легко понять то или иное. Надеюсь, что вы отнесетесь с пониманием.

Задача 11 из 11-ой лекции

Если вы написали через == то ничего страшного. Мне кажется здесь все и так понятно. Но если нет, то вы всегда можете задать вопрос в чате.

```
3 public static void main(String[] args) {
4     print(checkStrings( a: "one", b: "two"));
5     print(checkStrings( a: "", b: ""));
6     print(checkStrings( a: "", b: "two"));
7     print(checkStrings( a: "one", b: ""));
8     print(checkStrings( a: "one", b: "one"));
9 }
10
11 @ private static boolean checkStrings(String a, String b) {
12     return !a.isEmpty() && !b.isEmpty() && a.equals(b);
13 }
14
```

Main > main()

Run: Main x

/usr/lib/jvm/java-1.8.0-openjdk-amd64/bin/java ...

false
false
false
false
true

Process finished with exit code 0

Задача 12 из 11-ой лекции

```
3 public static void main(String[] args) {
4     print(checkIsUrl( text: "http://www.google.com"));
5     print(checkIsUrl( text: "www.google.com"));
6 }
7
8 @ private static boolean checkIsUrl(String text) {
9     return text.startsWith("http://");
10 }
11
```

Main > main()

Run: Main x

/usr/lib/jvm/java-1.8.0-openjdk-amd64/bin/java ...

true
false

Process finished with exit code 0

Здесь простые знания английского могли бы вам помочь. Строка начинается с – startsWith. Ничего сложного если ты немного знаешь английский и умеешь включить голову. Идем дальше!

Задача 13 из 11-ой лекции

```
3 public static void main(String[] args) {
4     checkMarks(5, 5, 5, 5, 5);
5     checkMarks(3, 5, 4, 2, 4);
6     checkMarks(3, 2, 4, 2, 3);
7     checkMarks(4, 4, 5, 4, 3);
8     checkMarks(4, 4, 5, 4, 4);
9 }
10
11 private static void checkMarks(int mark1, int mark2, int mark3, int mark4, int mark5) {
12     double average = (mark1 + mark2 + mark3 + mark4 + mark5) / 5.0;
13     if (average == 5) {
14         print(average + " Отличник");
15     } else if (average >= 4) {
16         print(average + " хорошист");
17     } else if (average >= 3) {
18         print(average + " троечник");
19     } else {
20         print(average + " двоечник");
21     }
22 }
23
24 private static void print(Object b) {
25     System.out.println(b);
26 }
27 }
```

Main \ checkMarks()

Run: Main x

/usr/lib/jvm/java-1.8.0-openjdk-amd64/bin/java ...

5.0 Отличник
3.6 троечник
2.8 двоечник
4.0 хорошист
4.2 хорошист

Здесь нужно было вспомнить, что если вы пытаетесь найти среднее арифметическое чисел, то вам нужно преобразовать `int` к `double`, а для этого нужно делить не на 5, а на 5.0. Можете сами проверить. Если будете делить на 5, то и получите `int` на самом деле, скажем так – сразу округленное значение.

Задача 14 из 11-ой лекции

Здесь я попробовал сделать получше. Смотрите. Сначала я проверю, есть ли среди имени или фамилии ключевое слово или нет. Если одно из двух есть, то я проверю на комбо. Если нет совпадения (&&) , то я выведу просто отказ. И пишем `else` для всех остальных случаев. Здесь конечно же надо было сделать 2 константы для Иван и Иванов. Ведь мы их используем 2 раза. Но мы позднее поговорим про `equals`. А сейчас пойдем дальше. И еще раз – неважно как вы написали код, если он в итоге рабочий. Сначала вы должны написать код, который правильно обрабатывает логику, для этого есть `мейн`, где можно просто вызвать функцию со всеми возможными параметрами и после уже улучшать рабочий код и каждый раз проверять – а не сломали ли вы что-то.

```
3 public static void main(String[] args) {
4     checkName( name: "Иван", surname: "Иванов");
5     checkName( name: "Иван", surname: "Сидоров");
6     checkName( name: "Николай", surname: "Иванов");
7     checkName( name: "Николай", surname: "Сидоров");
8 }
9
10 @ private static void checkName(String name, String surname) {
11     if (name.equals("Иван") || surname.equals("Иванов")) {
12         if (name.equals("Иван") && surname.equals("Иванов")) {
13             print("КОМБО!");
14         } else {
15             print("Отказ");
16         }
17     } else {
18         print("Здравствуйте, " + name + " " + surname);
19     }
20 }
```

Main > checkName()

Run: Main x

/usr/lib/jvm/java-1.8.0-openjdk-amd64/bin/java ...

КОМБО!

Отказ

Отказ

Здравствуйте, Николай Сидоров

Process finished with exit code 0

Задача 15 из 11-ой лекции

```
3 public static void main(String[] args) {
4     print(module( number: -5));
5     print(module( number: 6));
6     print(module( number: 0));
7 }
8
9 @ private static int module(int number) {
10     return number > 0 ? number : -1 * number;
11 }
12
```

Main > main()

Run: Main x

/usr/lib/jvm/java-1.8.0-openjdk-amd64/bin/java ...

5

6

0

Process finished with exit code 0

Обычным тернарным оператором сразу вернем положительное число или же умножим на -1.

Задача 16 из 11-ой лекции

```
3 public static void main(String[] args) {
4     isRectangle( AB: 5, BC: 5, CD: 5, DA: 5);
5     isRectangle( AB: 5, BC: 6, CD: 5, DA: 6);
6     isRectangle( AB: 5, BC: 4, CD: 5, DA: 6);
7 }
8
9 private static void isRectangle(int AB, int BC, int CD, int DA) {
10     if (AB == BC && BC == CD && CD == DA) {
11         print("квадрат");
12     } else if (AB == CD && BC == DA) {
13         print("прямоугольник");
14     } else {
15         print("четырёхугольник");
16     }
17 }
18
19 method void main(String[])
20 Main main()
```

Run: Main x

```
/usr/lib/jvm/java-1.8.0-openjdk-amd64/bin/java ...
квадрат
прямоугольник
четырёхугольник
Process finished with exit code 0
```

Я облегчил себе жизнь тем, что требую на вход стороны четырехугольника в правильной последовательности. И еще я не проверил значения на ноль и отрицательные. Но думаю вы это сделаете сами. Вы уже знаете как. Конечно же у нас может быть ромб вместо квадрата и параллелограмм вместо прямоугольника, ведь мы не знаем ничего про углы. Но мы опустили этот момент. Можете выводить нечто типа – квадрат или ромб. В этой задаче главным было использовать логические операторы. Кстати вы можете написать код иначе. Например сравнить пары на равенство и если после этого они равны меж собой то это квадрат. Получится вложенный if.

```
private static void isRectangle(int AB, int BC, int CD, int DA) {
    if (AB == CD && BC == DA) {
        if (AB == BC) {
            print("квадрат");
        } else {
            print("прямоугольник");
        }
    } else {
        print("четырёхугольник");
    }
}
```


Задача 17 из 11-ой лекции

```
3 public static void main(String[] args) {
4     print(isLessonUnlocked( lessonNumber: 1,  isUserPremium: false));
5     print(isLessonUnlocked( lessonNumber: 12, isUserPremium: true));
6     print(isLessonUnlocked( lessonNumber: 1,  isUserPremium: true));
7     print(isLessonUnlocked( lessonNumber: 13, isUserPremium: false));
8     print(isLessonUnlocked( lessonNumber: 103, isUserPremium: true));
9 }
10
11 private static boolean isLessonUnlocked(int lessonNumber, boolean isUserPremium) {
12     if (lessonNumber < 1 || lessonNumber > 100) {
13         print("Номер урока невалидный");
14         return false;
15     } else {
16         return isUserPremium || lessonNumber == 1;
17     }
18 }
19
```

Main > isLessonUnlocked()

Run: Main x

/usr/lib/jvm/java-1.8.0-openjdk-amd64/bin/java ...

true
true
true
false
Номер урока невалидный
false

Process finished with exit code 0

Опять же, сначала проверка входных данных и только после уже смотрим на логику. Если премиум юзер, то неважно какой урок, а если урок номер 1, то неважно какой юзер. Значит можно использовать логическое или.

Задача 18 из 11-ой лекции

```
1 public class Main {
2
3     private static boolean playerNumber1Now = true;
4
5     public static void main(String[] args) {
6         doStep( playerNumber: 1, step: "step1");
7         doStep( playerNumber: 1, step: "step2");
8         doStep( playerNumber: 2, step: "step3");
9         doStep( playerNumber: 2, step: "step4");
10        doStep( playerNumber: 1, step: "step5");
11    }
12
13    private static void doStep(int playerNumber, String step) {
14        if (playerNumber == 1 && playerNumber1Now ||
15            playerNumber == 2 && !playerNumber1Now) {
16            print(step);
17            playerNumber1Now = !playerNumber1Now;
18        }
19    }
20 }
```

Main

Run: Main x

/usr/lib/jvm/java-1.8.0-openjdk-amd64/bin/java ...

step1
step3
step5

Process finished with exit code 0

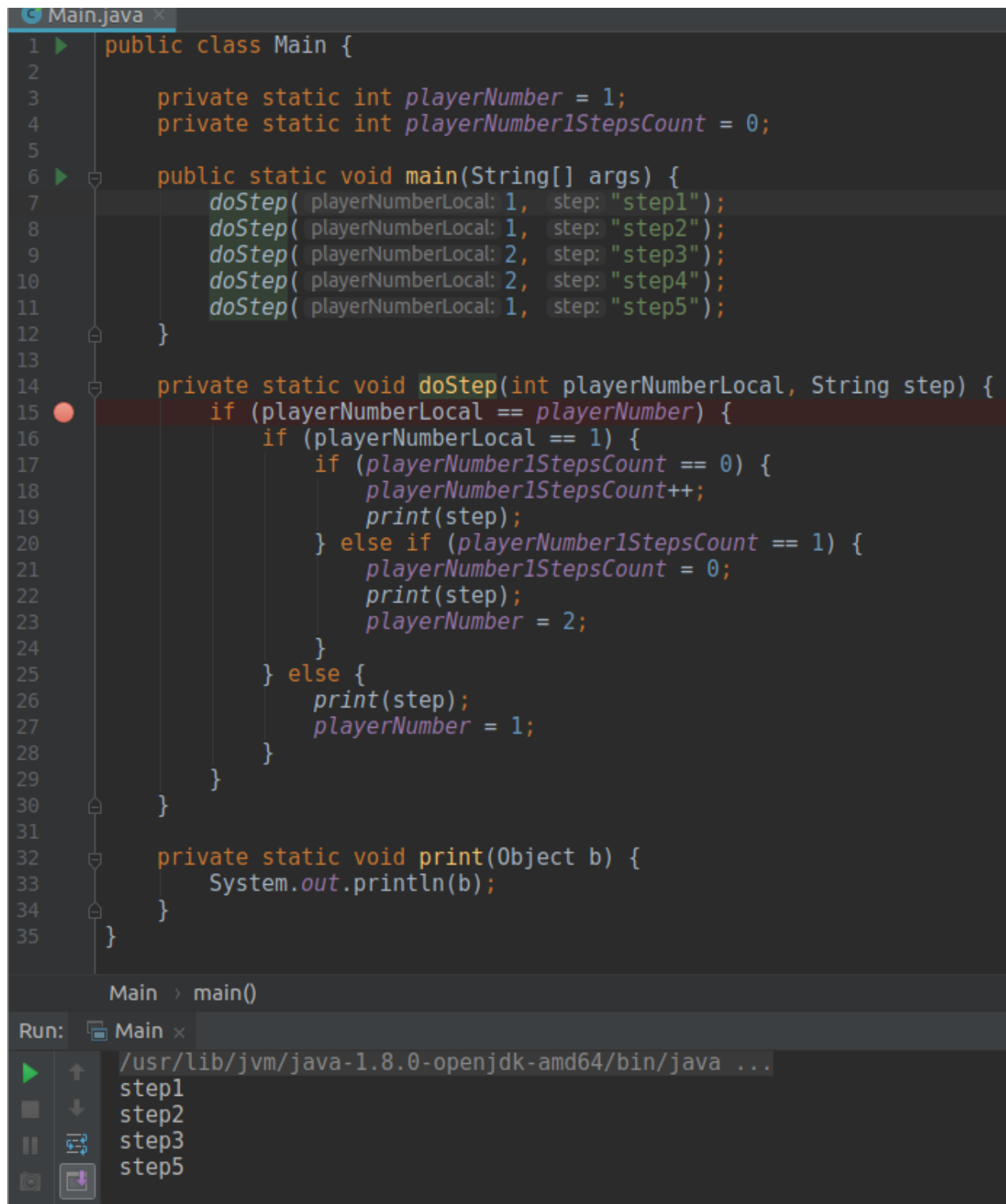
Я написал так, но можно было бы и использовать `int` переменную для хранения номера игрока и тогда бы было проще.

```
2
3     private static int playerNumber = 1;
4
5     public static void main(String[] args) {
6         doStep( playerNumberLocal: 1, step: "step1");
7         doStep( playerNumberLocal: 1, step: "step2");
8         doStep( playerNumberLocal: 2, step: "step3");
9         doStep( playerNumberLocal: 2, step: "step4");
10        doStep( playerNumberLocal: 1, step: "step5");
11    }
12
13    private static void doStep(int playerNumberLocal, String step) {
14        if (playerNumberLocal == playerNumber) {
15            print(step);
16            playerNumber = playerNumber == 1 ? 2 : 1;
17        }
18    }
19 }
```

Я сравниваю номер игрока с номером игрока текущего. Если они совпадают, то я делаю шаг. И после того, как я сделал шаг, я меняю текущего игрока. Если раньше это была цифра 1, то теперь станет 2, иначе 1 (т.е. иначе если не было равно 1, то есть сейчас 2, то станет 1).

Как видите тернарные операторы очень помогают нам. Посмотрите на линию 17 в предыдущем скриншоте, там я меняю значение true на false и обратно. А здесь цифру 1 на 2 и обратно. Но и в этом случае кода меньше просто. Условие проще читается.

Задача 19 из 11-ой лекции



```
1 public class Main {
2
3     private static int playerNumber = 1;
4     private static int playerNumberStepsCount = 0;
5
6     public static void main(String[] args) {
7         doStep( playerNumberLocal: 1, step: "step1");
8         doStep( playerNumberLocal: 1, step: "step2");
9         doStep( playerNumberLocal: 2, step: "step3");
10        doStep( playerNumberLocal: 2, step: "step4");
11        doStep( playerNumberLocal: 1, step: "step5");
12    }
13
14    private static void doStep(int playerNumberLocal, String step) {
15        if (playerNumberLocal == playerNumber) {
16            if (playerNumberLocal == 1) {
17                if (playerNumberStepsCount == 0) {
18                    playerNumberStepsCount++;
19                    print(step);
20                } else if (playerNumberStepsCount == 1) {
21                    playerNumberStepsCount = 0;
22                    print(step);
23                    playerNumber = 2;
24                }
25            } else {
26                print(step);
27                playerNumber = 1;
28            }
29        }
30    }
31
32    private static void print(Object b) {
33        System.out.println(b);
34    }
35 }
```

Run: Main x /usr/lib/jvm/java-1.8.0-openjdk-amd64/bin/java ...

step1
step2
step3
step5

Здесь немного посложнее. И если вам нелегко сразу написать красивый код, то просто напишите такой, который работает. Я сам написал сначала этот. И я мог бы его переписать иначе, но вот вам самое наивное главное правило – РАБОТАЕТ – НЕ ТРОГАЙ.

Ведь действительно, если ваш код работает, то зачем его трогать? Ведь можно сломать. А как узнать, что ты сломал? Для этого нужно чтобы у вас была проверка. И мы пишем проверку метода в мейне со всеми возможными аргументами и комбинациями.

Итак, в конце этой лекции, которая больше практика, я бы хотел сказать вот что:

когда я учился программированию, то днями не мог решить какую-то задачу и я бросал это дело не один раз, но каждый раз через некоторое время я предпринимал еще одну попытку понять и сделать. Так что и вы, друзья мои, не отчаивайтесь, если у вас не получается что-то сразу. Бывает, что ты вроде написал код, а понял его суть лишь спустя время. Так что просто мой совет : идите дальше, главное не останавливаться. Я так в 2013 году забросил идею стать программистом после первой неудачной попытки найти работу и из моей жизни почти 3 года ушло вникуда. Да, я попробовал опять лишь спустя почти 3 года (по факту 2) и претерпевал неудачи много раз перед тем как удача улыбнулась мне.

Не вешаем нос и учимся дальше. Вы всегда можете вернуться к задачам и попробовать выполнить их самостоятельно.