|  |  |  |
| --- | --- | --- |
| Министерство науки и высшего образования РФ  Федеральное государственное автономное образовательное учреждение высшего образования  «Пермский государственный национальный исследовательский университет» | | |
|  | Институт компьютерных наук и технологий | |
| **ОТЧЁТ**  по лабораторной работе №1  по дисциплине «Языки программирования»  Вариант 2 | | |
|  | | Работу выполнил  студент группы ДВБ7-ПР3-2023 НБ 2 курса  Румянцев В.И.  «14» октября 2024 г. |
| Работу проверил  Ракина В.Д.  «\_\_\_» \_\_\_\_\_\_\_\_\_ 2024 г. |
| Пермь 2024 | | |

СОДЕРЖАНИЕ

[Задание 1.2. 6](#_Toc181051455)

[Текст задания 6](#_Toc181051456)

[Алгоритм решения 6](#_Toc181051457)

[Тестирование 6](#_Toc181051458)

[Код программы 7](#_Toc181051459)

[Задание 1.4. 8](#_Toc181051460)

[Текст задания 8](#_Toc181051461)

[Алгоритм решения 8](#_Toc181051462)

[Тестирование 8](#_Toc181051463)

[Код программы 9](#_Toc181051464)

[Задание 1.6. 10](#_Toc181051465)

[Текст задания 10](#_Toc181051466)

[Алгоритм решения 10](#_Toc181051467)

[Тестирование 10](#_Toc181051468)

[Код программы 11](#_Toc181051469)

[Задание 1.8. 12](#_Toc181051470)

[Текст задания 12](#_Toc181051471)

[Алгоритм решения 12](#_Toc181051472)

[Тестирование 12](#_Toc181051473)

[Код программы 13](#_Toc181051474)

[Задание 1.10. 14](#_Toc181051475)

[Текст задания 14](#_Toc181051476)

[Алгоритм решения 14](#_Toc181051477)

[Тестирование 15](#_Toc181051478)

[Код программы 16](#_Toc181051479)

[Задание 2.2. 17](#_Toc181051480)

[Текст задания 17](#_Toc181051481)

[Алгоритм решения 17](#_Toc181051482)

[Тестирование 17](#_Toc181051483)

[Код программы 18](#_Toc181051484)

[Задание 2.4. 19](#_Toc181051485)

[Текст задания 19](#_Toc181051486)

[Алгоритм решения 19](#_Toc181051487)

[Тестирование 19](#_Toc181051488)

[Код программы 20](#_Toc181051489)

[Задание 2.6. 21](#_Toc181051490)

[Текст задания 21](#_Toc181051491)

[Алгоритм решения 21](#_Toc181051492)

[Тестирование 21](#_Toc181051493)

[Код программы 22](#_Toc181051494)

[Задание 2.8. 23](#_Toc181051495)

[Текст задания 23](#_Toc181051496)

[Алгоритм решения 23](#_Toc181051497)

[Тестирование 23](#_Toc181051498)

[Код программы 24](#_Toc181051499)

[Задание 2.10. 25](#_Toc181051500)

[Текст задания 25](#_Toc181051501)

[Алгоритм решения 25](#_Toc181051502)

[Тестирование 26](#_Toc181051503)

[Код программы 27](#_Toc181051504)

[Задание 3.2. 28](#_Toc181051505)

[Текст задания 28](#_Toc181051506)

[Алгоритм решения 28](#_Toc181051507)

[Тестирование 28](#_Toc181051508)

[Код программы 29](#_Toc181051509)

[Задание 3.4. 30](#_Toc181051510)

[Текст задания 30](#_Toc181051511)

[Алгоритм решения 30](#_Toc181051512)

[Тестирование 30](#_Toc181051513)

[Код программы 31](#_Toc181051514)

[Задание 3.6. 32](#_Toc181051515)

[Текст задания 32](#_Toc181051516)

[Алгоритм решения 32](#_Toc181051517)

[Тестирование 32](#_Toc181051518)

[Код программы 33](#_Toc181051519)

[Задание 3.8. 34](#_Toc181051520)

[Текст задания 34](#_Toc181051521)

[Алгоритм решения 34](#_Toc181051522)

[Тестирование 34](#_Toc181051523)

[Код программы 34](#_Toc181051524)

[Задание 3.10. 35](#_Toc181051525)

[Текст задания 35](#_Toc181051526)

[Алгоритм решения 35](#_Toc181051527)

[Тестирование 36](#_Toc181051528)

[Код программы 36](#_Toc181051529)

[Задание 4.2. 37](#_Toc181051530)

[Текст задания 37](#_Toc181051531)

[Алгоритм решения 37](#_Toc181051532)

[Тестирование 37](#_Toc181051533)

[Код программы 37](#_Toc181051534)

[Задание 4.4. 38](#_Toc181051535)

[Текст задания 38](#_Toc181051536)

[Алгоритм решения 38](#_Toc181051537)

[Тестирование 38](#_Toc181051538)

[Код программы 39](#_Toc181051539)

[Задание 4.6. 40](#_Toc181051540)

[Текст задания 40](#_Toc181051541)

[Алгоритм решения 40](#_Toc181051542)

[Тестирование 40](#_Toc181051543)

[Код программы 41](#_Toc181051544)

[Задание 4.8. 42](#_Toc181051545)

[Текст задания 42](#_Toc181051546)

[Алгоритм решения 42](#_Toc181051547)

[Тестирование 43](#_Toc181051548)

[Код программы 43](#_Toc181051549)

[Задание 4.10. 44](#_Toc181051550)

[Текст задания 44](#_Toc181051551)

[Алгоритм решения 44](#_Toc181051552)

[Тестирование 44](#_Toc181051553)

[Код программы 45](#_Toc181051554)

# Задание 1.2.

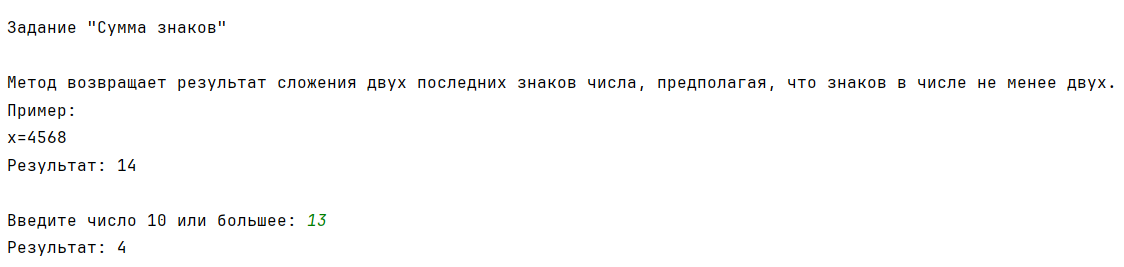
## Текст задания

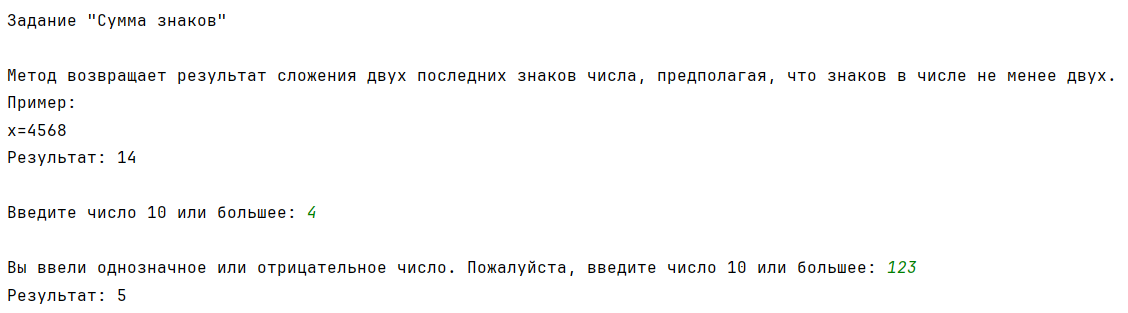
1.2. Сумма знаков. Дана сигнатура метода: public int sumLastNums (int x); Необходимо реализовать метод таким образом, чтобы он возвращал результат сложения двух последних знаков числах, предполагая, что знаков в числе не менее двух. Подсказки: int x=123%10; // х будет иметь значение 3 int у=123/10; // у будет иметь значение 12 Пример: x=4568 результат: 14

## Алгоритм решения

1. Получение последнего знака числа
2. Получение предпоследнего знака числа
3. Сложение последнего и предпоследнего знаков
4. Возврат результата

## Тестирование





## Код программы

https://github.com/Vladimir-Rumyantsev/JavaLabs/blob/master/src/LabWork1/Main.java

# Задание 1.4.

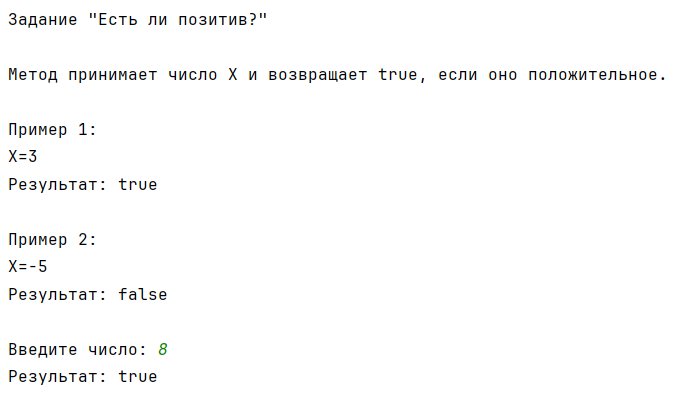
## Текст задания

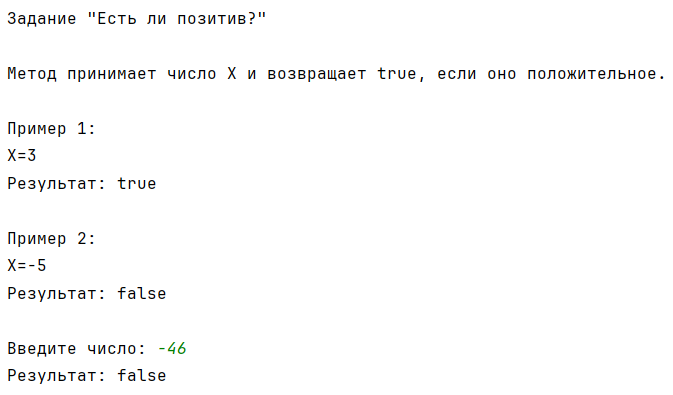
Есть ли позитив. Дана сигнатура метода: public bool isPositive (int x); Необходимо реализовать метод таким образом, чтобы он принимал число x и возвращал true, если оно положительное. Пример 1: x=3 результат: true Пример 2: x=-5 результат: false

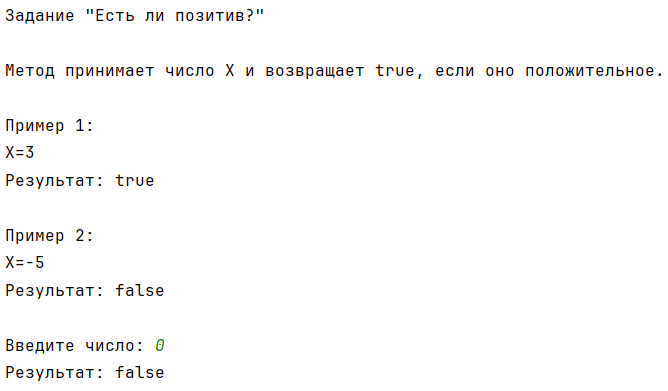
## Алгоритм решения

1. Проверка знака числа
2. Возврат true, если число положительное
3. Возврат false, если число не положительное

## Тестирование







## Код программы

https://github.com/Vladimir-Rumyantsev/JavaLabs/blob/master/src/LabWork1/Main.java

# Задание 1.6.

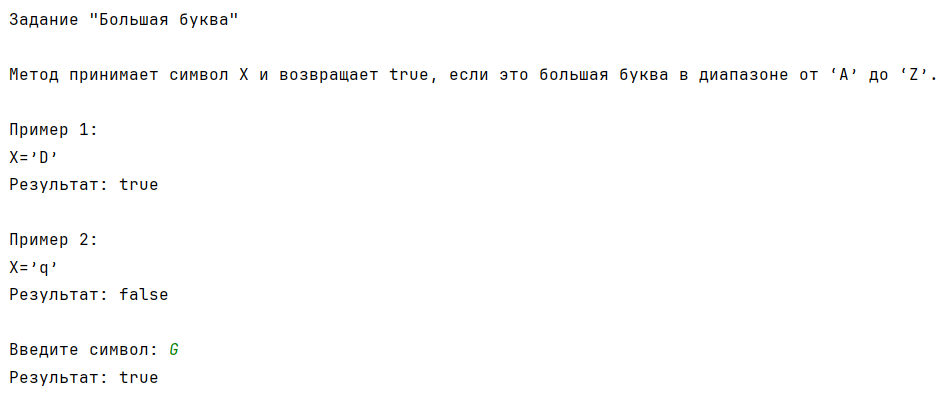
## Текст задания

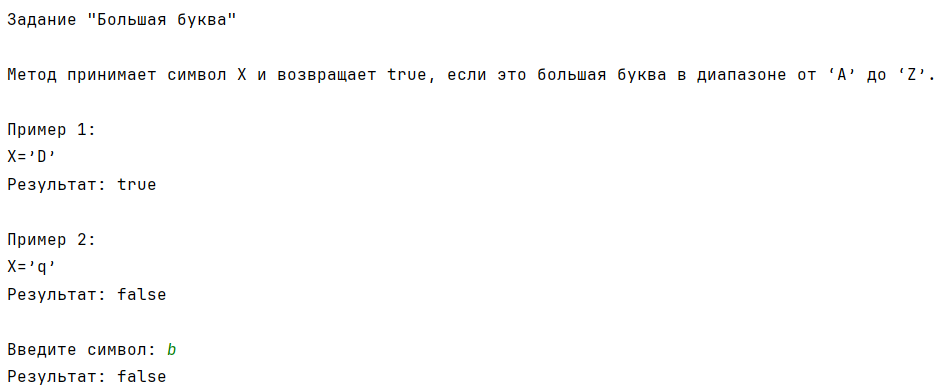
Большая буква. Дана сигнатура метода: public bool isUpperCase (char x); Необходимо реализовать метод таким образом, чтобы он принимал символ x и возвращал true, если это большая буква в диапазоне от ‘A’ до ‘Z’. Пример 1: x=’D’ результат: true Пример 2: x=’q’ результат: false

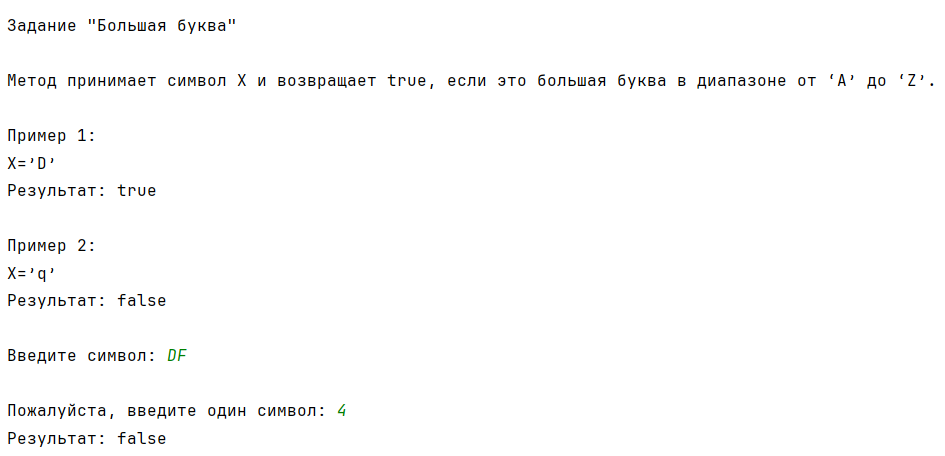
## Алгоритм решения

1. Проверка, находится ли символ в диапазоне от 'A' до 'Z'
2. Возврат true, если символ является большой буквой
3. Возврат false, если символ не является большой буквой

## Тестирование







## Код программы

https://github.com/Vladimir-Rumyantsev/JavaLabs/blob/master/src/LabWork1/Main.java

# Задание 1.8.

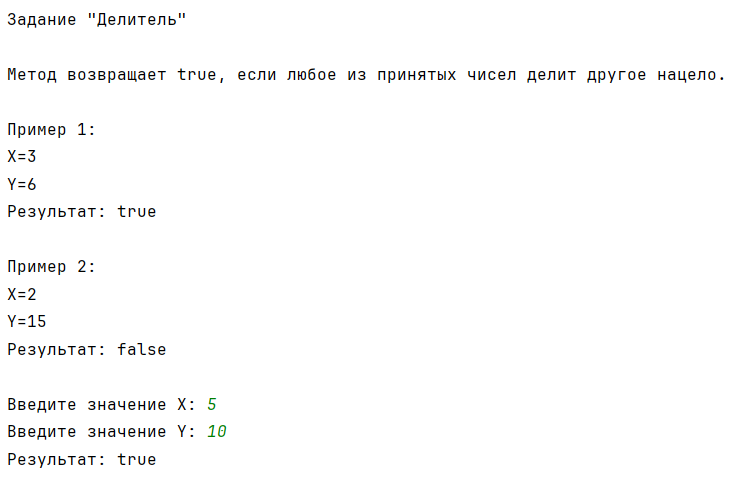
## Текст задания

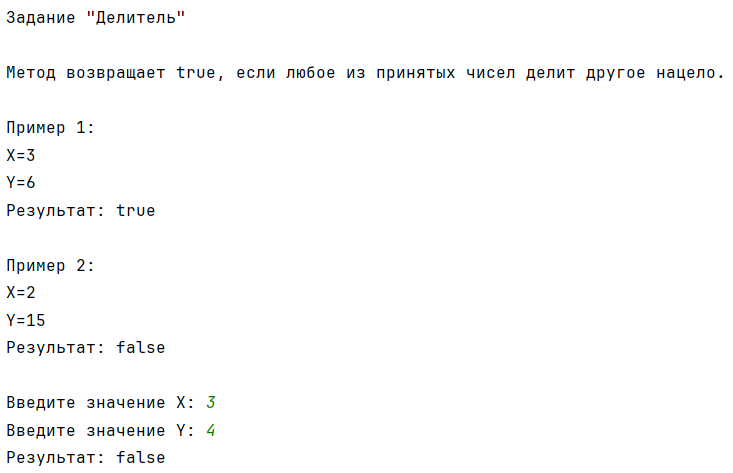
Делитель. Дана сигнатура метода: public bool isDivisor (int a, int b); Необходимо реализовать метод таким образом, чтобы он возвращал true, если любое из принятых чисел делит другое нацело. Пример 1: a=3 b=6 результат: true Пример 2: a=2 b=15 результат: false

## Алгоритм решения

1. Проверка, делится ли число a на число b нацело
2. Проверка, делится ли число b на число a нацело
3. Возврат true, если хотя бы одно из условий выполняется
4. Возврат false, если ни одно из условий не выполняется

## Тестирование





## Код программы

https://github.com/Vladimir-Rumyantsev/JavaLabs/blob/master/src/LabWork1/Main.java

# Задание 1.10.

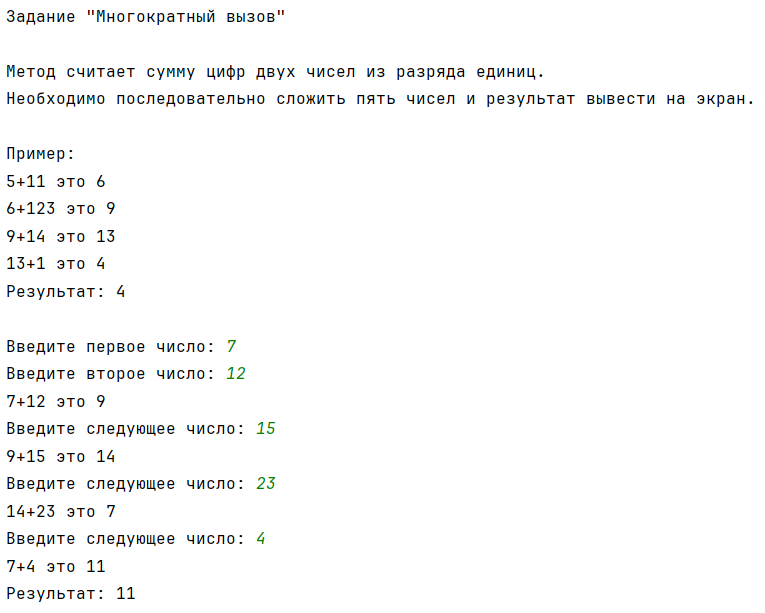
## Текст задания

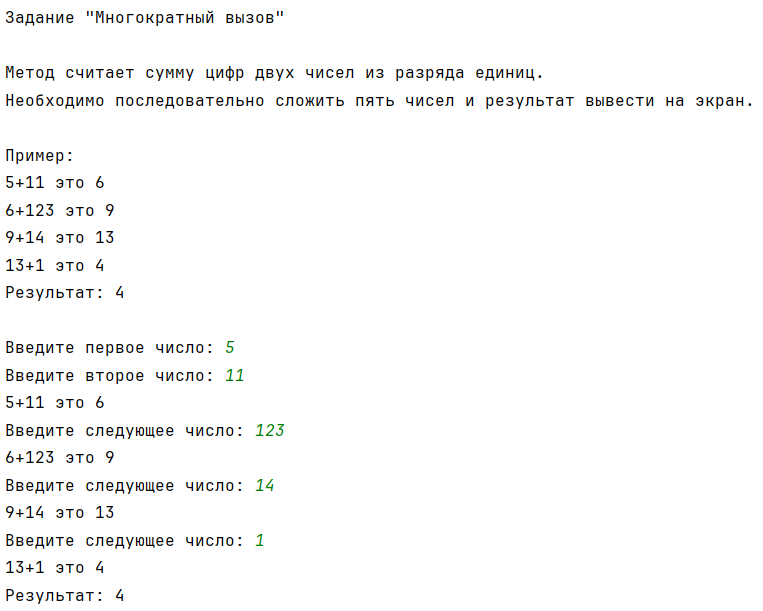
Многократный вызов. Дана сигнатура метода: public int lastNumSum(int a, int b) Необходимо реализовать метод таким образом, чтобы он считал сумму цифр двух чисел из разряда единиц. Выполните с его помощью последовательное сложение пяти чисел и результат выведите на экран. Постарайтесь выполнить задачу, используя минимально возможное количество вспомогательных переменных. Пример: 5+11 это 6 6+123 это 9 9+14 это 13 13+1 это 4 Итого 4

## Алгоритм решения

1. Получение последнего знака числа a
2. Получение последнего знака числа b
3. Сложение последних знаков чисел a и b
4. Возврат результата сложения
5. Повторение шагов 1-4 для последовательного сложения пяти чисел
6. Вывод итогового результата на экран

## Тестирование





## Код программы

https://github.com/Vladimir-Rumyantsev/JavaLabs/blob/master/src/LabWork1/Main.java

# Задание 2.2.

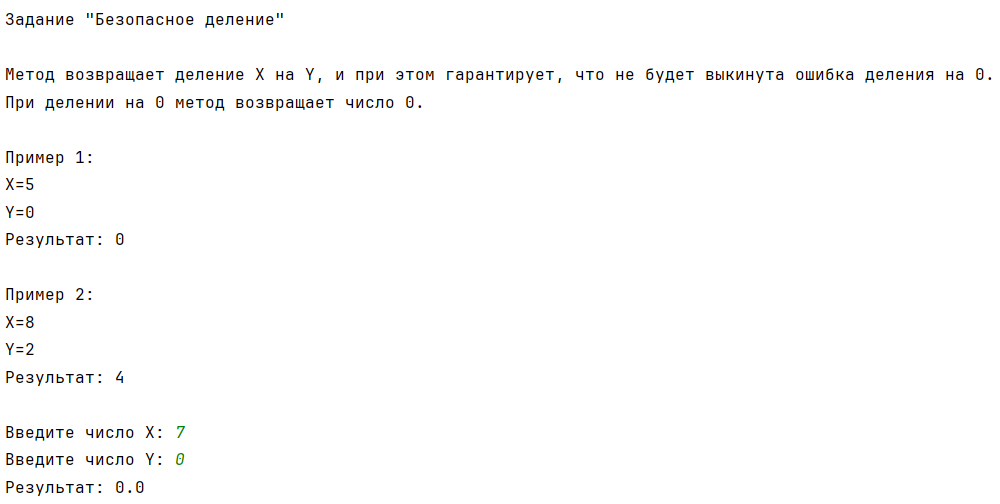
## Текст задания

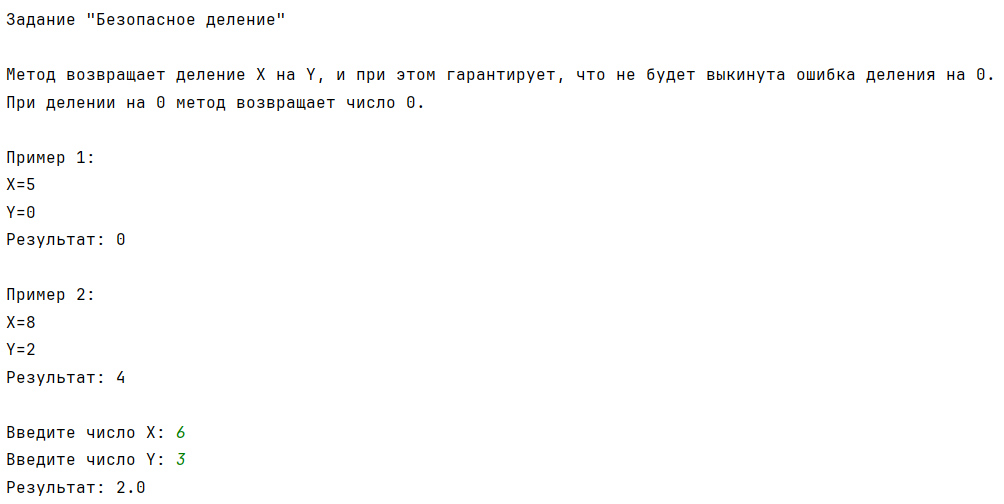
Безопасное деление. Дана сигнатура метода: public double safeDiv (int x, int y); Необходимо реализовать метод таким образом, чтобы он возвращал деление x на y, и при этом гарантировал, что не будет выкинута ошибка деления на 0. При делении на 0 следует вернуть из метода число 0. Подсказка: смотри ограничения на операции типов данных. Пример 1: x=5 y=0 результат: 0 Пример 2: x=8 y=2 результат: 4

## Алгоритм решения

1. Проверка, равен ли делитель (y) нулю
2. Возврат 0, если делитель равен нулю
3. Преобразование делимого (x) в тип double
4. Выполнение деления делимого на делитель
5. Возврат результата деления

## Тестирование





## Код программы

https://github.com/Vladimir-Rumyantsev/JavaLabs/blob/master/src/LabWork1/Main.java

# Задание 2.4.

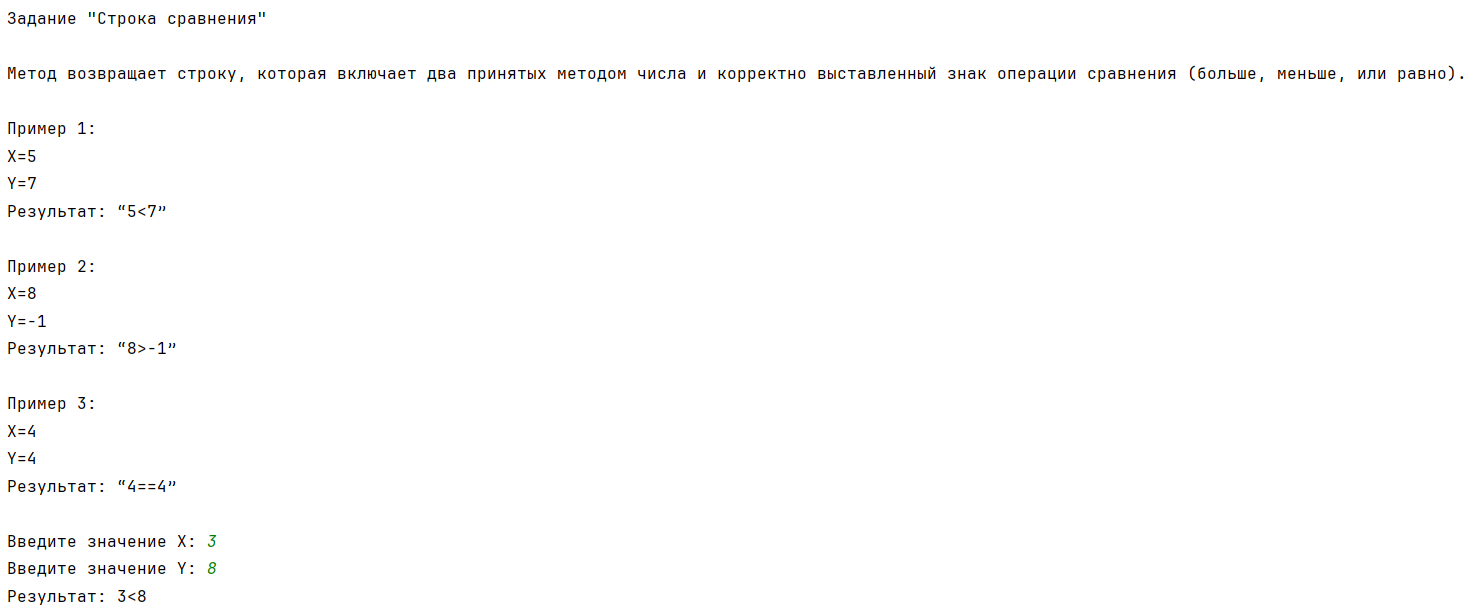
## Текст задания

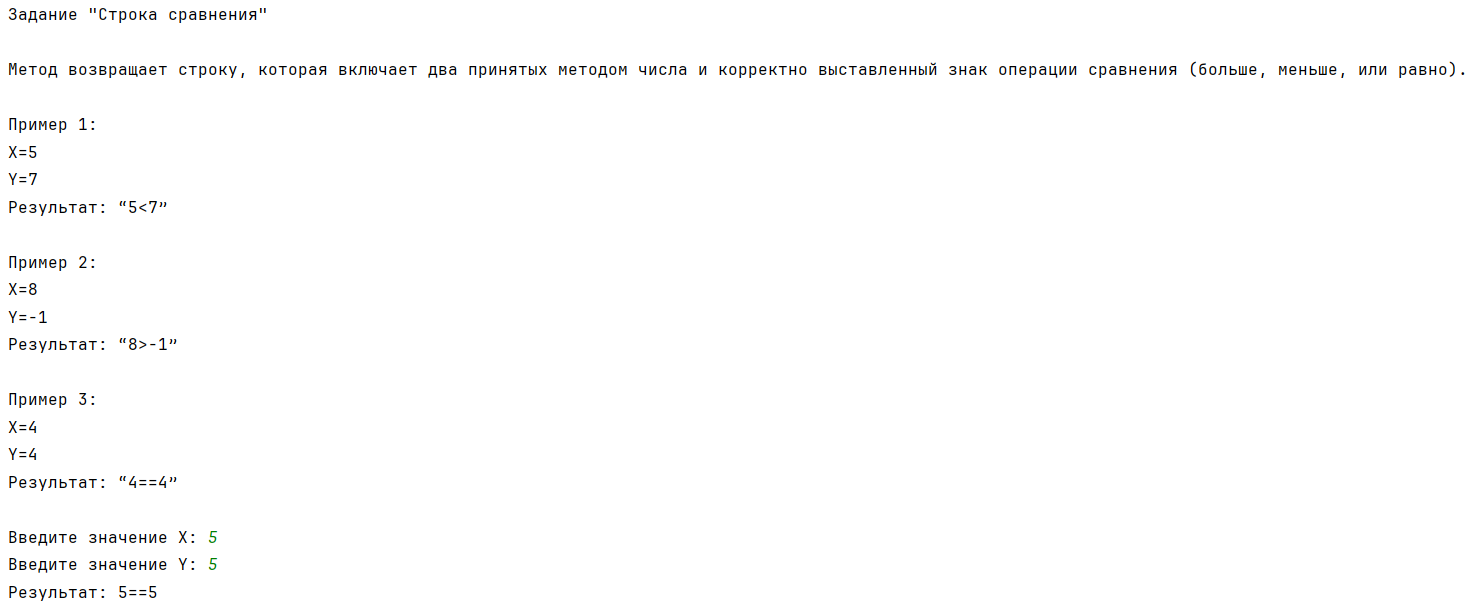
Строка сравнения. Дана сигнатура метода: public String makeDecision (int x, int y); Необходимо реализовать метод таким образом, чтобы он возвращал строку, которая включает два принятых методом числа и корректно выставленный знак операции сравнения (больше, меньше, или равно). Пример 1: x=5 y=7 результат: “5< 7” Пример 2: x=8 y=-1 результат: “8 >-1” Пример 3: x=4 y=4 результат: “4==4”

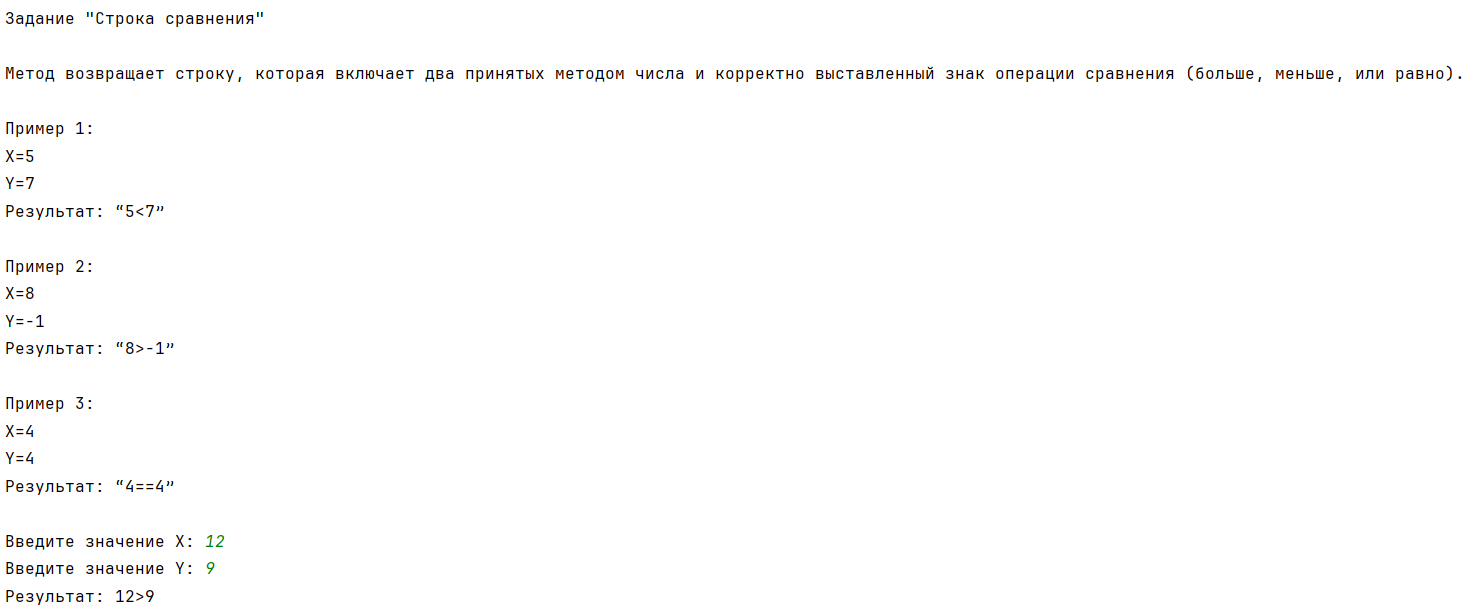
## Алгоритм решения

1. Сравнение чисел x и y
2. Возврат строки с оператором ">" если x больше y
3. Возврат строки с оператором "<" если x меньше y
4. Возврат строки с оператором "==" если x равно y

## Тестирование







## Код программы

https://github.com/Vladimir-Rumyantsev/JavaLabs/blob/master/src/LabWork1/Main.java

# Задание 2.6.

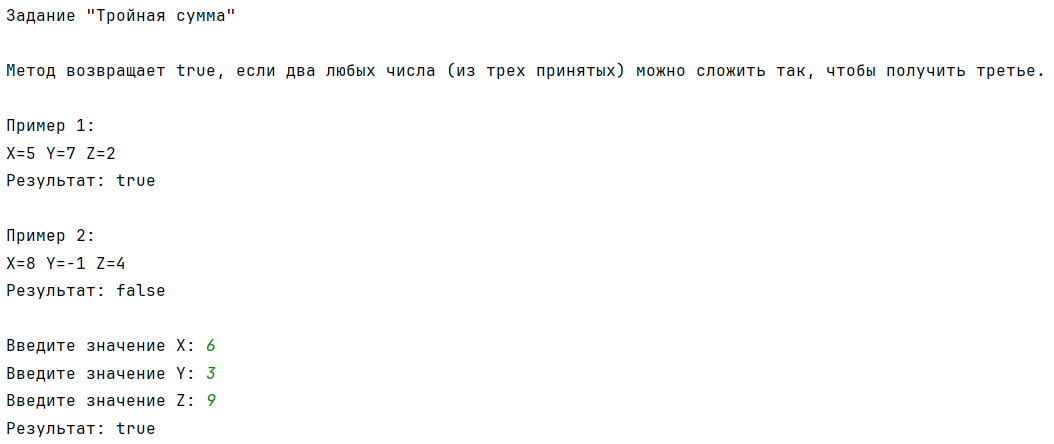
## Текст задания

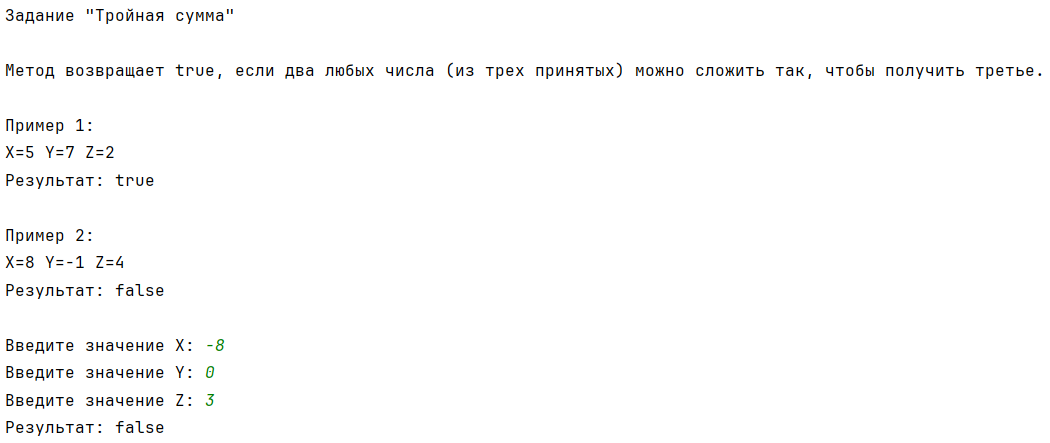
Тройная сумма. Дана сигнатура метода: public bool sum3 (int x, int y, int z); Необходимо реализовать метод таким образом, чтобы он возвращал true, если два любых числа (из трех принятых) можно сложить так, чтобы получить третье. Пример 1: x=5 y=7 z=2 результат: true Пример 2: x=8 y=-1 z=4 результат: false

## Алгоритм решения

1. Проверка, равна ли сумма x и y числу z
2. Проверка, равна ли сумма x и z числу y
3. Проверка, равна ли сумма y и z числу x
4. Возврат true, если хотя бы одно из условий выполняется
5. Возврат false, если ни одно из условий не выполняется

## Тестирование





## Код программы

https://github.com/Vladimir-Rumyantsev/JavaLabs/blob/master/src/LabWork1/Main.java

# Задание 2.8.

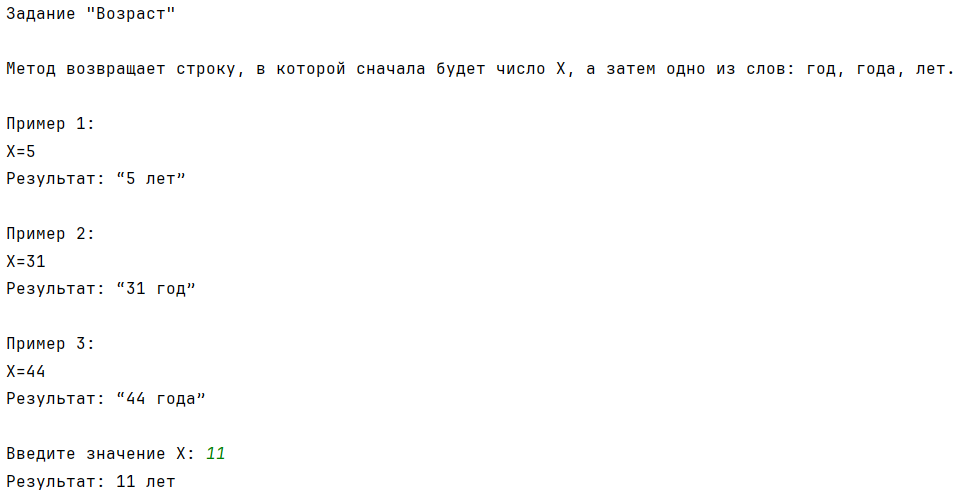
## Текст задания

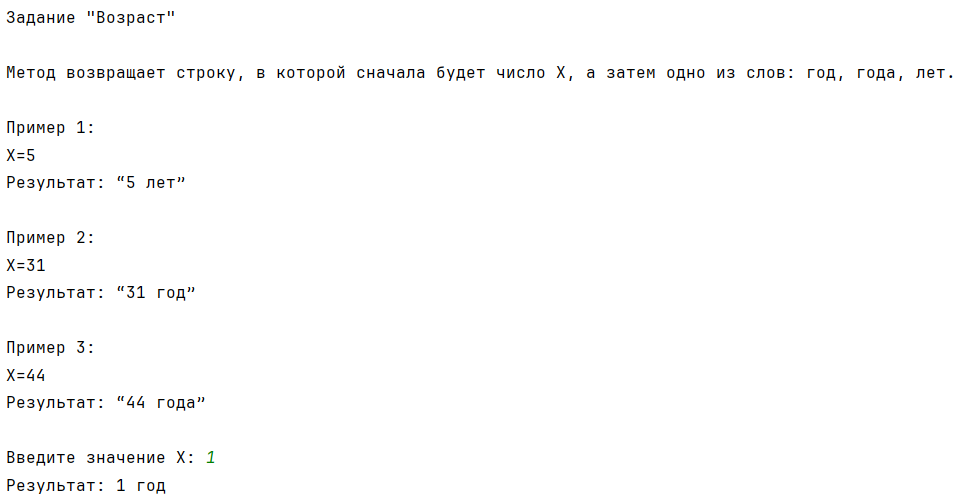
Возраст. Дана сигнатура метода: public String age (int x); Необходимо реализовать метод таким образом, чтобы он возвращал строку, в которой сначала будет число х, а затем одно из слов: год года лет Слово “год” добавляется, если число х заканчивается на 1, кроме числа 11. Слово “года” добавляется, если число х заканчивается на 2, 3 или 4, кроме чисел 12, 13, 14. Слово “лет”добавляется во всех остальных случаях. Подсказка: оператор % позволяет получить остаток от деления. Пример 1: x=5 результат: “5 лет” Пример 2: x=31 результат: “31 год” Пример 3: x=44 результат: “44 года”

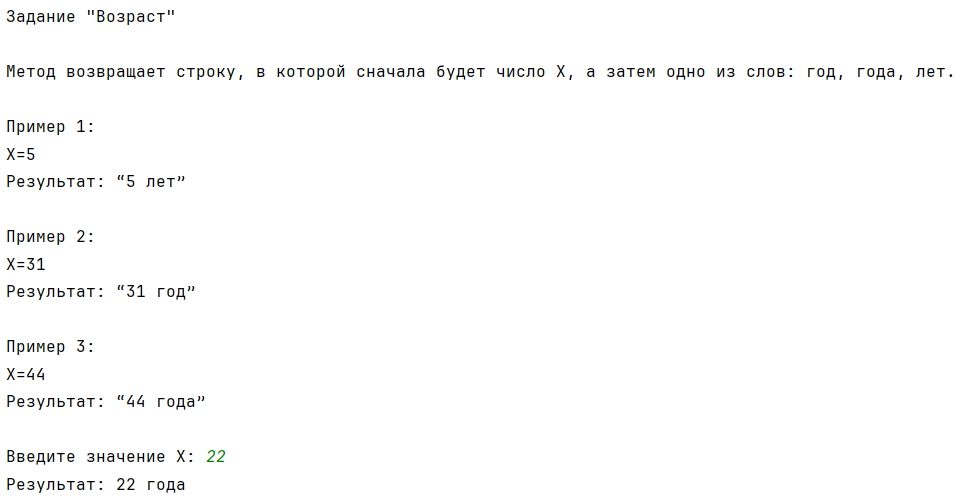
## Алгоритм решения

1. Проверка, заканчивается ли число на 0, 5, 6, 7, 8, 9, 11, 12, 13 или 14
2. Возврат строки с числом x и словом "лет", если условие выполняется
3. Проверка, заканчивается ли число x на 1
4. Возврат строки с числом x и словом "год", если условие выполняется
5. Возврат строки с числом x и словом "года" во всех остальных случаях

## Тестирование







## Код программы

https://github.com/Vladimir-Rumyantsev/JavaLabs/blob/master/src/LabWork1/Main.java

# Задание 2.10.

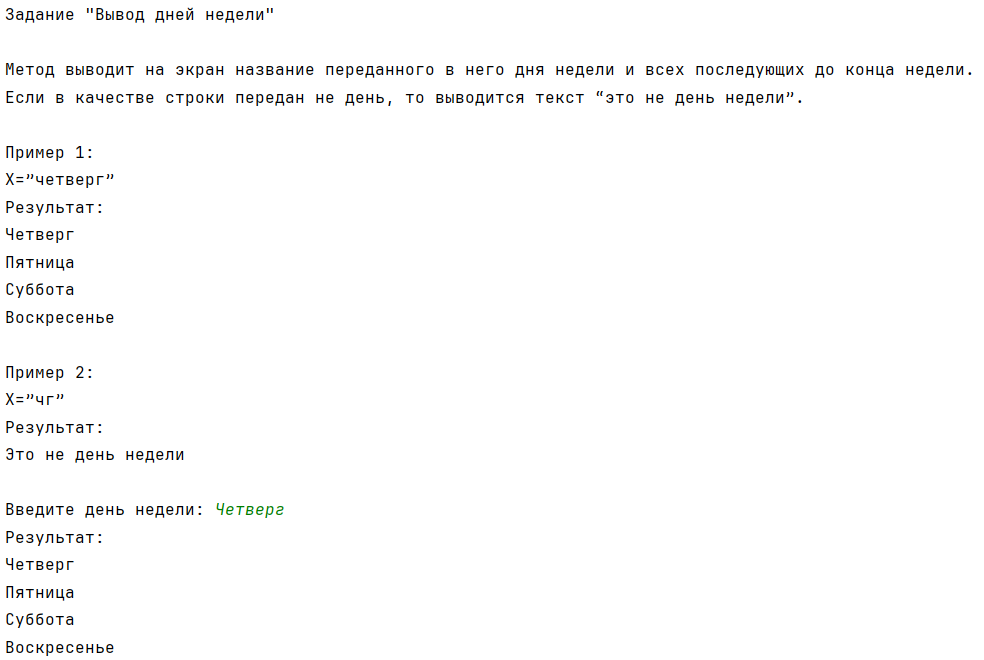
## Текст задания

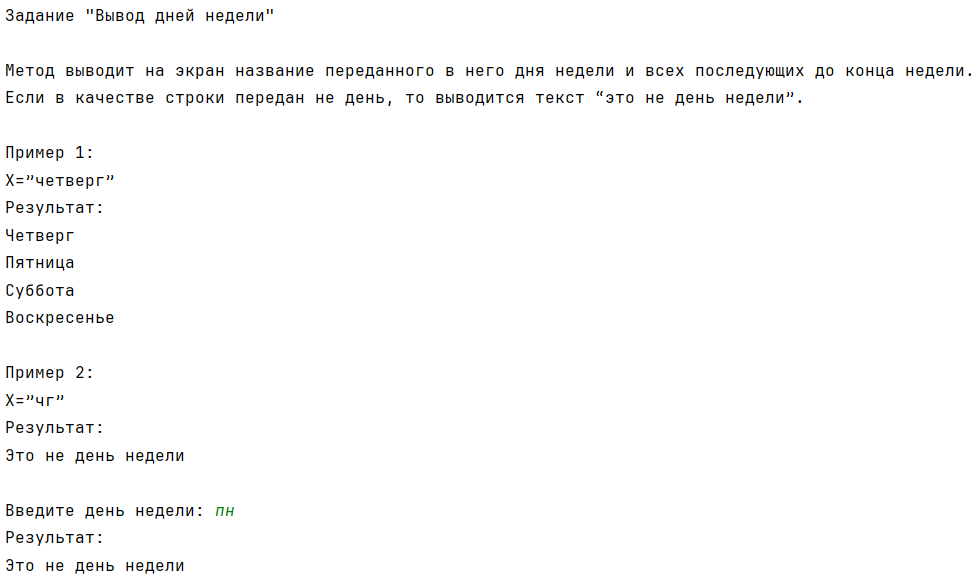
Вывод дней недели. Дана сигнатура метода: public void printDays (String x); В качестве параметра метод принимает строку, в которой записано название дня недели. Необходимо реализовать метод таким образом, чтобы он выводил на экран название переданного в него дня и всех последующих до конца недели дней. Если в качестве строки передан не день, то выводится текст “это не день недели”. Первый день понедельник, последний – воскресенье. Вместо if в данной задаче используйте switch. Пример 1: x=”четверг” результат: четверг пятница суббота воскресенье Пример 2: x=”чг” результат: это не день недели

## Алгоритм решения

1. Преобразование первого символа строки x в верхний регистр, остальные символы в нижний регистр
2. Использование оператора switch для проверки значения строки x
3. Вывод на экран названия переданного дня и всех последующих дней недели до конца
4. Если значение строки x не является днем недели, вывод текста "Это не день недели"

## Тестирование





## Код программы

https://github.com/Vladimir-Rumyantsev/JavaLabs/blob/master/src/LabWork1/Main.java

# Задание 3.2.

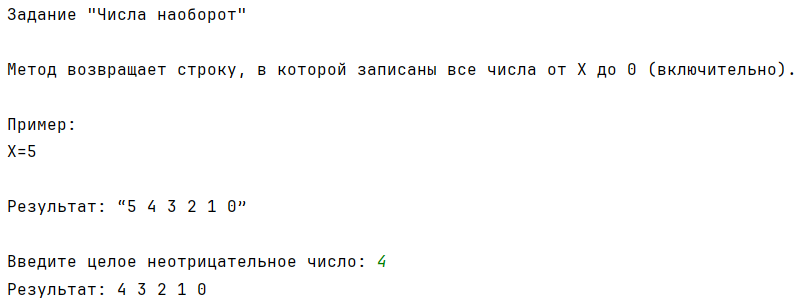
## Текст задания

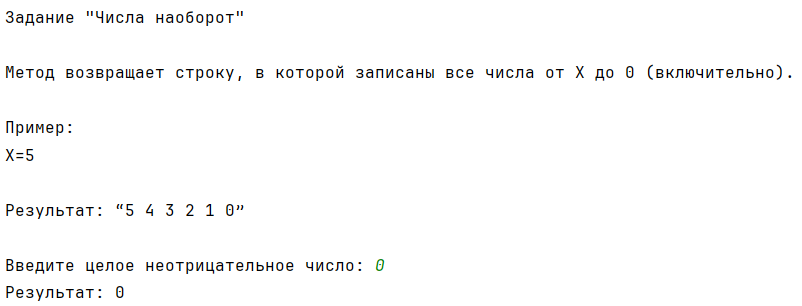
Числа наоборот. Дана сигнатура метода: public String reverseListNums (int x); Необходимо реализовать метод таким образом, чтобы он возвращал строку, в которой будут записаны все числа от x до 0 (включительно). Пример: x=5 результат: “5 4 3 2 1 0”

## Алгоритм решения

1. Создание объекта StringBuilder для формирования строки
2. Использование цикла while для итерации от x до 0 включительно
3. Добавление текущего значения x в строку с пробелом перед ним
4. Уменьшение значения x на 1
5. Возврат сформированной строки

## Тестирование





## Код программы

https://github.com/Vladimir-Rumyantsev/JavaLabs/blob/master/src/LabWork1/Main.java

# Задание 3.4.

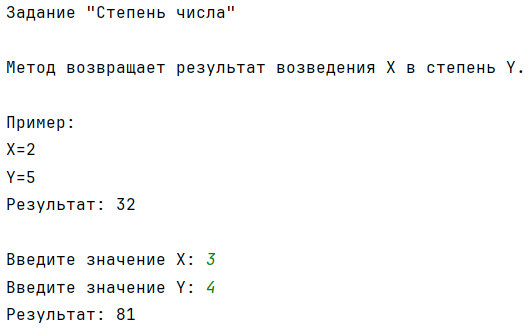
## Текст задания

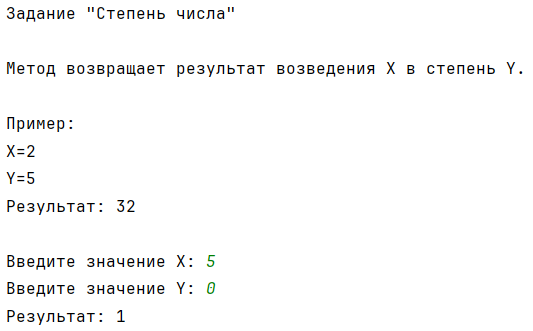
Степень числа. Дана сигнатура метода: public int pow (int x, int y); Необходимо реализовать метод таким образом, чтобы он возвращал результат возведения x в степень y. Подсказка: для получения степени необходимо умножить единицу на число x, и сделать это y раз, т.е. два в третьей степени это 1\*2\*2\*2 Пример: x=2 y=5 результат: 32

## Алгоритм решения

1. Инициализация переменной для хранения результата значением 1
2. Использование цикла for для умножения переменной на x, y раз
3. Возврат результата

## Тестирование





## Код программы

https://github.com/Vladimir-Rumyantsev/JavaLabs/blob/master/src/LabWork1/Main.java

# Задание 3.6.

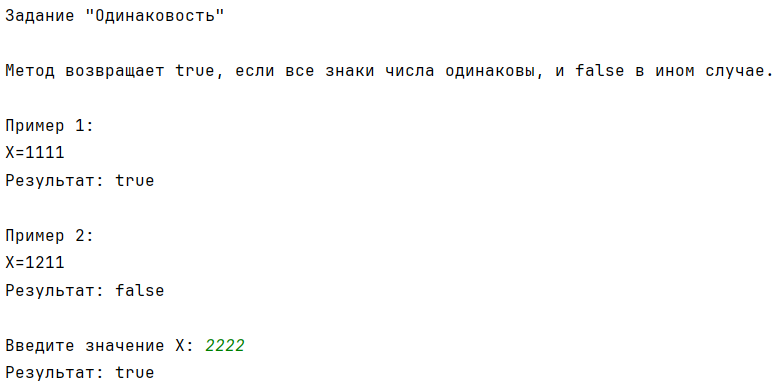
## Текст задания

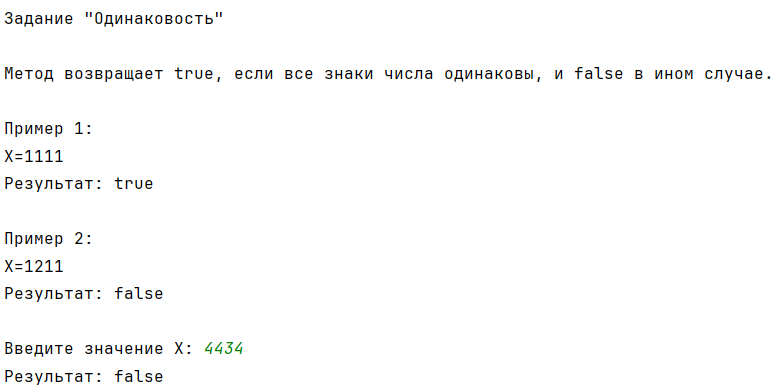
Одинаковость. Дана сигнатура метода: public bool equalNum (int x); Необходимо реализовать метод таким образом, чтобы он возвращал true, если все знаки числа одинаковы, и false в ином случае. Подсказки: intx=123%10; // х будет иметь значение 3 intу=123/10; // у будет иметь значение 12 Пример 1: x=1111 результат: true Пример 2: x=1211 результат: false

## Алгоритм решения

1. Получение последнего знака числа x
2. Удаление последнего знака числа x
3. Использование цикла while для проверки оставшихся знаков числа x
4. Сравнение текущего знака с последним знаком
5. Возврат false, если найден знак, отличный от последнего
6. Возврат true, если все знаки одинаковы

## Тестирование





## Код программы

https://github.com/Vladimir-Rumyantsev/JavaLabs/blob/master/src/LabWork1/Main.java

# Задание 3.8.

## Текст задания

Левый треугольник. Дана сигнатура метода: public void leftTriangle (int x); Необходимо реализовать метод таким образом, чтобы он выводил на экран треугольник из символов ‘\*’ у которого х символов в высоту, а количество символов в ряду совпадает с номером строки. Пример 1: x=2 результат: \* \*\* Пример 2: x=4 результат: \* \*\* \*\*\* \*\*\*\*

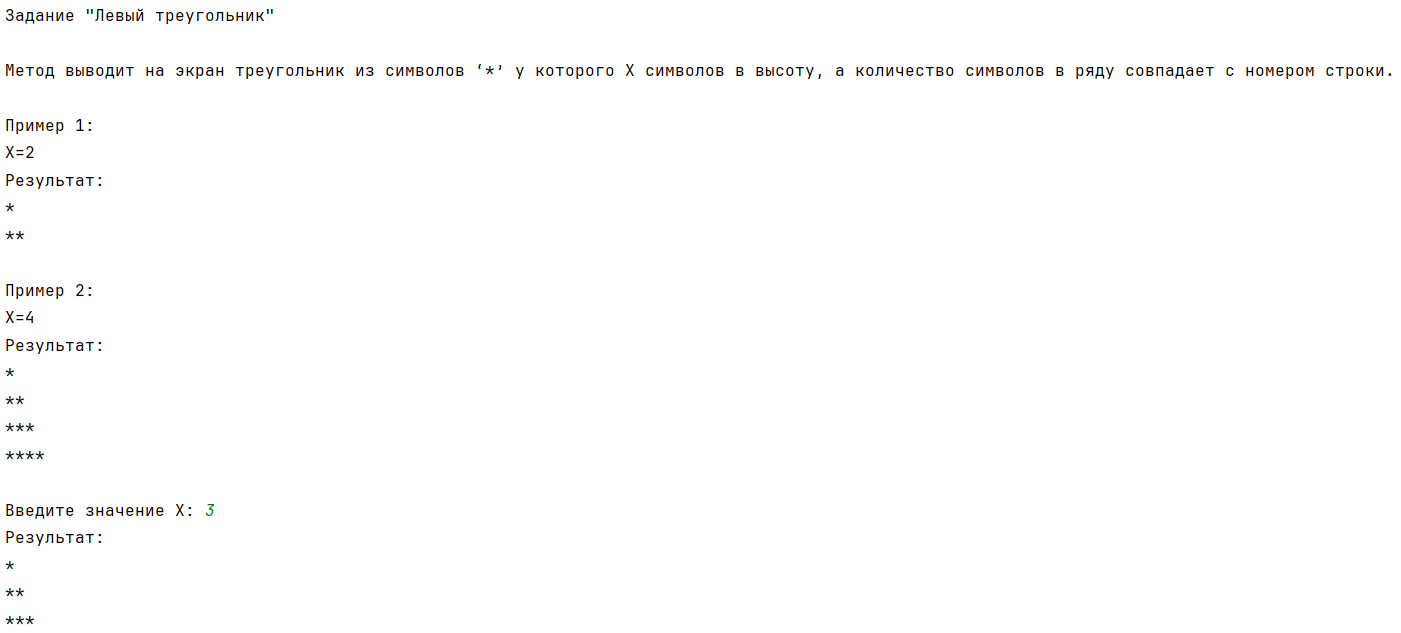
## Алгоритм решения

Использование цикла for для итерации от 1 до x включительно

В каждой итерации вывод строки, содержащей i символов '\*'

Переход на новую строку после вывода каждой строки

## Тестирование



## Код программы

https://github.com/Vladimir-Rumyantsev/JavaLabs/blob/master/src/LabWork1/Main.java

# Задание 3.10.

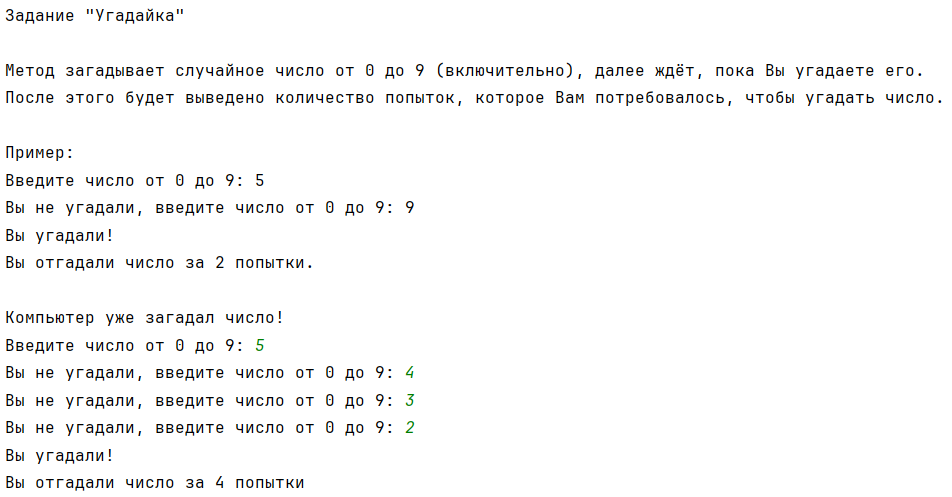
## Текст задания

Угадайка. Дана сигнатура метода: public void guessGame() Необходимо реализовать метод таким образом, чтобы он генерировал случайное число от 0 до 9, далее считывал с консоли введенное пользователем число и выводил, угадал ли пользователь то, что было загадано, или нет. Метод запускается до тех пор, пока пользователь не угадает число. После этого выведите на экран количество попыток, которое потребовалось пользователю, чтобы угадать число. Пример: Введите число от 0 до 9: 5 Вы не угадали, введите число от 0 до 9: 9 Вы угадали! Вы отгадали число за 2 попытки

## Алгоритм решения

1. Генерация случайного числа от 0 до 9
2. Запрос у пользователя ввода числа от 0 до 9
3. Инициализация счетчика попыток значением 1
4. Использование цикла while для проверки, угадал ли пользователь число
5. Увеличение счетчика попыток на 1 при каждой неудачной попытке
6. Запрос у пользователя повторного ввода числа при неудачной попытке
7. Проверка количества попыток для корректного вывода слова "попытка", "попытки" или "попыток"
8. Вывод сообщения о количестве попыток, потребовавшихся для угадывания числа

## Тестирование



## Код программы

https://github.com/Vladimir-Rumyantsev/JavaLabs/blob/master/src/LabWork1/Main.java

# Задание 4.2.

## Текст задания

Поиск последнего значения. Дана сигнатура метода: public int findLast (int[] arr, int x); Необходимо реализовать метод таким образом, чтобы он возвращал индекс последнего вхождения числа x в массив arr. Если число не входит в массив – возвращается -1. Пример: arr=[1,2,3,4,2,2,5] x=2 результат: 5

## Алгоритм решения

1. Инициализация цикла for для итерации по массиву arr с конца к началу
2. Проверка, равен ли текущий элемент массива числу x
3. Возврат индекса текущего элемента, если он равен x
4. Возврат -1, если число x не найдено в массиве

## Тестирование



## Код программы

https://github.com/Vladimir-Rumyantsev/JavaLabs/blob/master/src/LabWork1/Main.java

# Задание 4.4.

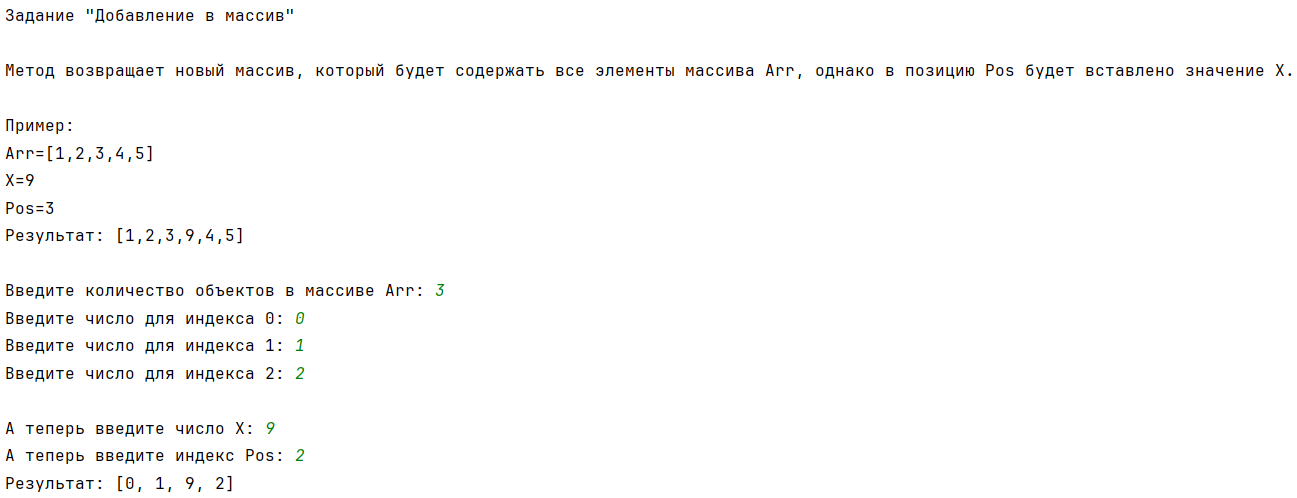
## Текст задания

Добавление в массив. Дана сигнатура метода: public int[]add (int[] arr, int x, int pos); Необходимо реализовать метод таким образом, чтобы он возвращал новый массив, который будет содержать все элементы массива arr, однако в позицию pos будет вставлено значение x. Пример: arr=[1,2,3,4,5] x=9 pos=3 результат: [1,2,3,9,4,5]

## Алгоритм решения

1. Определение длины исходного массива arr
2. Проверка и корректировка значения pos, если оно выходит за пределы допустимого диапазона
3. Создание нового массива newArr, длина которого на 1 больше длины исходного массива
4. Копирование элементов из исходного массива в новый массив до позиции pos
5. Вставка значения x в позицию pos нового массива
6. Копирование оставшихся элементов из исходного массива в новый массив после позиции pos
7. Возврат нового массива

## Тестирование



## Код программы

https://github.com/Vladimir-Rumyantsev/JavaLabs/blob/master/src/LabWork1/Main.java

# Задание 4.6.

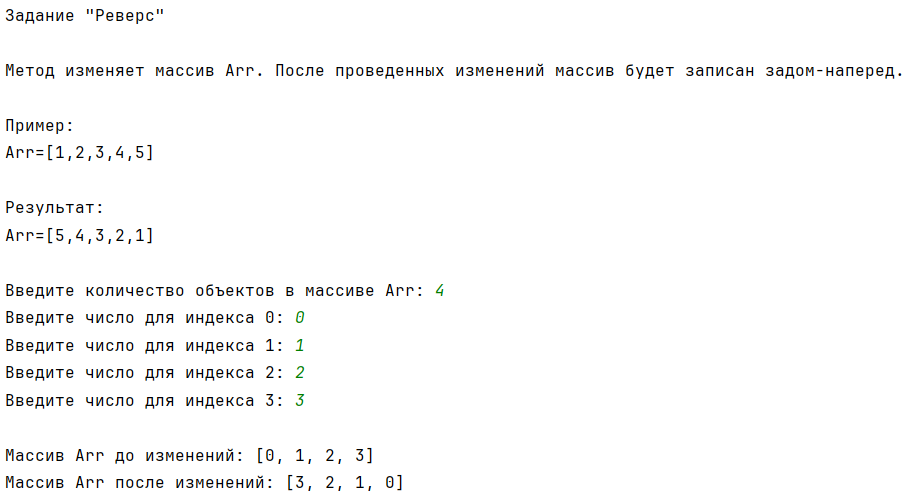
## Текст задания

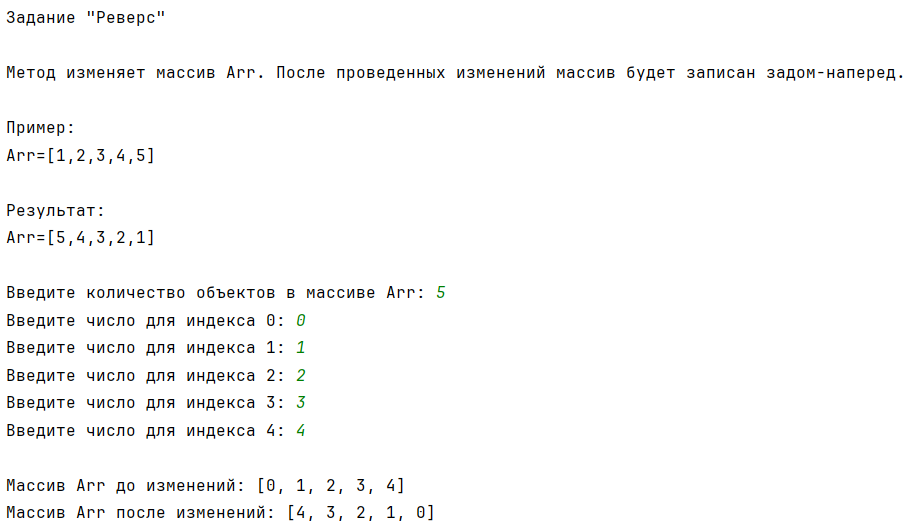
Реверс. Дана сигнатура метода: public void reverse (int[] arr); Необходимо реализовать метод таким образом, чтобы он изменял массив arr. После проведенных изменений массив должен быть записан задом-наперед. Пример: arr=[1,2,3,4,5] результат: arr=[5,4,3,2,1]

## Алгоритм решения

1. Инициализация двух указателей: left на начало массива и right на конец массива
2. Использование цикла while для обмена элементов массива, пока left меньше right
3. Обмен элементов на позициях left и right
4. Увеличение указателя left и уменьшение указателя right
5. Повторение шагов 3-4 до тех пор, пока left не станет больше или равен right

## Тестирование





## Код программы

https://github.com/Vladimir-Rumyantsev/JavaLabs/blob/master/src/LabWork1/Main.java

# Задание 4.8.

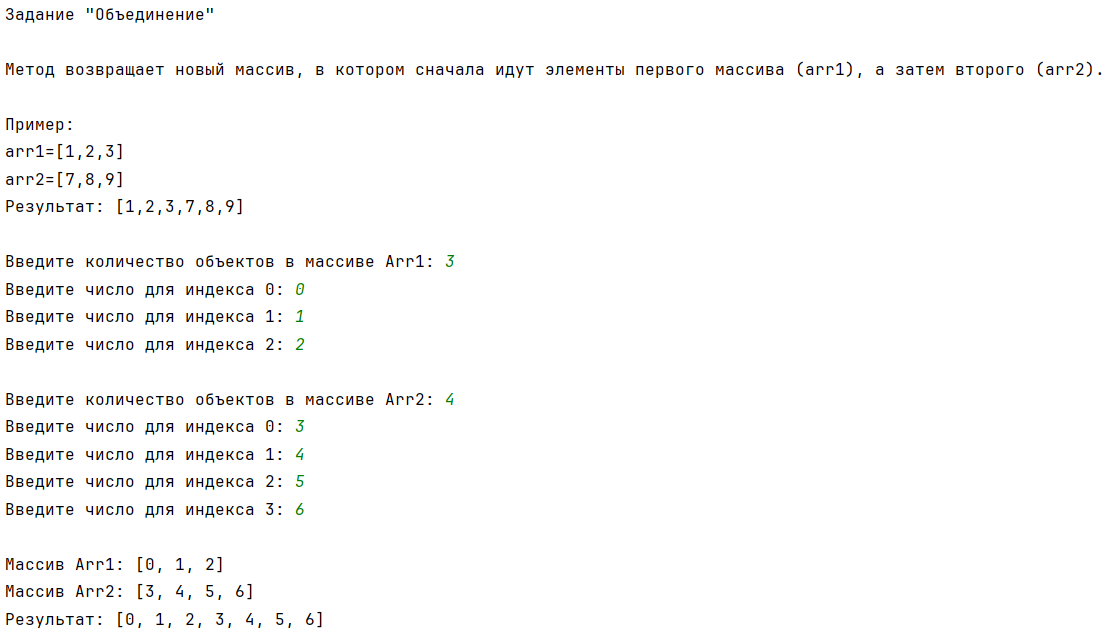
## Текст задания

Объединение. Дана сигнатура метода: public int[] concat (int[] arr1,int[] arr2); Необходимо реализовать метод таким образом, чтобы он возвращал новый массив, в котором сначала идут элементы первого массива (arr1), а затем второго (arr2). Пример: arr1=[1,2,3] arr2=[7,8,9] результат: [1,2,3,7,8,9]

## Алгоритм решения

1. Определение длины первого массива arr1
2. Определение длины второго массива arr2
3. Создание нового массива newArr, длина которого равна сумме длин первого и второго массивов
4. Копирование элементов из первого массива arr1 в новый массив newArr
5. Копирование элементов из второго массива arr2 в новый массив newArr, начиная с позиции после последнего элемента первого массива
6. Возврат нового массива newArr

## Тестирование



## Код программы

https://github.com/Vladimir-Rumyantsev/JavaLabs/blob/master/src/LabWork1/Main.java

# Задание 4.10.

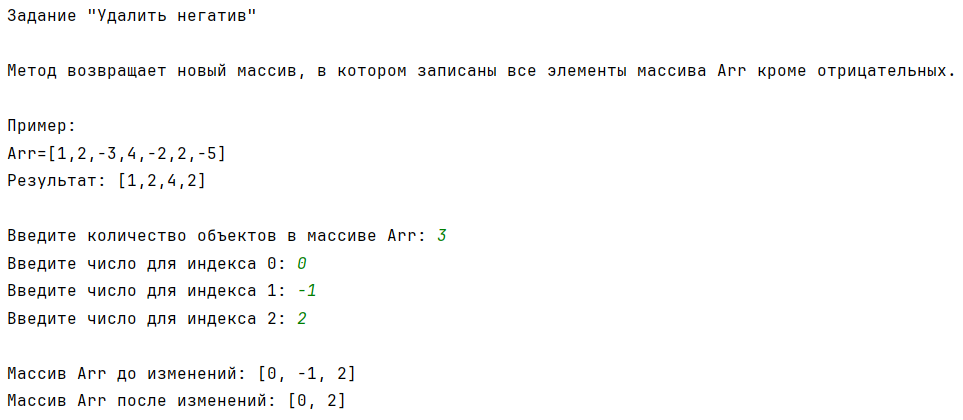
## Текст задания

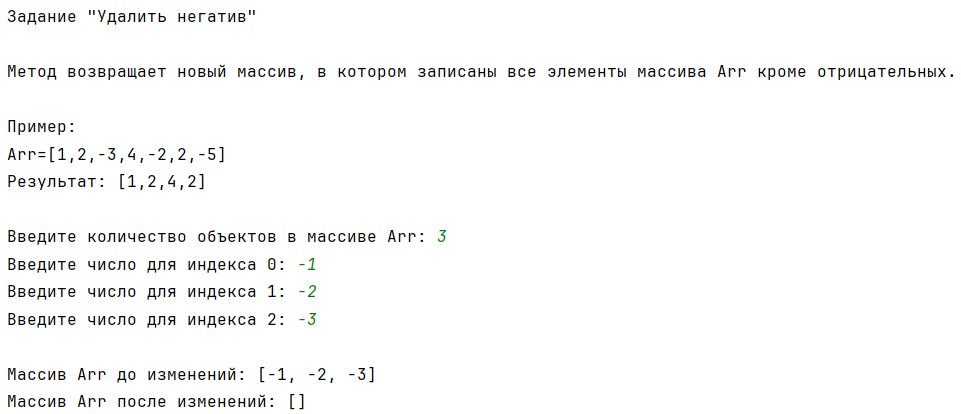
Удалить негатив. Дана сигнатура метода: public int[] deleteNegative (int[] arr); Необходимо реализовать метод таким образом, чтобы он возвращал новый массив, в котором записаны все элементы массива arr кроме отрицательных. Пример: arr=[1,2,-3,4,-2,2,-5] результат: [1,2,4,2]

## Алгоритм решения

1. Подсчет количества неотрицательных элементов в массиве arr
2. Создание нового массива newArr, длина которого равна количеству неотрицательных элементов
3. Копирование неотрицательных элементов из массива arr в новый массив newArr
4. Возврат нового массива newArr

## Тестирование





## Код программы

https://github.com/Vladimir-Rumyantsev/JavaLabs/blob/master/src/LabWork1/Main.java