|  |  |  |
| --- | --- | --- |
| 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 ПМИ-1,2-2023 2 курса  Хисматов Е.С.  «07» октября 2024 г. |
| Работу проверил  Ракина В. Д.  «\_\_\_» \_\_\_\_\_\_\_\_\_ 2024 г. |
| Пермь 2024 | | |

СОДЕРЖАНИЕ

[Задание 1.2 6](#_Toc179236994)

[Текст задания 6](#_Toc179236995)

[Алгоритм решения 6](#_Toc179236996)

[Тестирование 6](#_Toc179236997)

[Код программы 6](#_Toc179236998)

[Задание 1.4 7](#_Toc179236999)

[Текст задания 7](#_Toc179237000)

[Алгоритм решения 7](#_Toc179237001)

[Тестирование 7](#_Toc179237002)

[Код программы 8](#_Toc179237003)

[Задание 1.6 9](#_Toc179237004)

[Текст задания 9](#_Toc179237005)

[Алгоритм решения 9](#_Toc179237006)

[Тестирование 9](#_Toc179237007)

[Код программы 9](#_Toc179237008)

[Задание 1.8 10](#_Toc179237009)

[Текст задания 10](#_Toc179237010)

[Алгоритм решения 10](#_Toc179237011)

[Тестирование 10](#_Toc179237012)

[Код программы 10](#_Toc179237013)

[Задание 1.10 11](#_Toc179237014)

[Текст задания 11](#_Toc179237015)

[Алгоритм решения 11](#_Toc179237016)

[Тестирование 11](#_Toc179237017)

[Код программы 12](#_Toc179237018)

[Задание 2.2 13](#_Toc179237019)

[Текст задания 13](#_Toc179237020)

[Алгоритм решения 13](#_Toc179237021)

[Тестирование 13](#_Toc179237022)

[Код программы 13](#_Toc179237023)

[Задание 2.4 14](#_Toc179237024)

[Текст задания 14](#_Toc179237025)

[Алгоритм решения 14](#_Toc179237026)

[Тестирование 14](#_Toc179237027)

[Код программы 14](#_Toc179237028)

[Задание 2.6 15](#_Toc179237029)

[Текст задания 15](#_Toc179237030)

[Алгоритм решения 15](#_Toc179237031)

[Тестирование 15](#_Toc179237032)

[Код программы 15](#_Toc179237033)

[Задание 2.8 16](#_Toc179237034)

[Текст задания 16](#_Toc179237035)

[Алгоритм решения 16](#_Toc179237036)

[Тестирование 16](#_Toc179237037)

[Код программы 16](#_Toc179237038)

[Задание 2.10 17](#_Toc179237039)

[Текст задания 17](#_Toc179237040)

[Алгоритм решения 17](#_Toc179237041)

[Тестирование 17](#_Toc179237042)

[Код программы 17](#_Toc179237043)

[Задание 3.2 18](#_Toc179237044)

[Текст задания 18](#_Toc179237045)

[Алгоритм решения 18](#_Toc179237046)

[Тестирование 18](#_Toc179237047)

[Код программы 18](#_Toc179237048)

[Задание 3.4 19](#_Toc179237049)

[Текст задания 19](#_Toc179237050)

[Алгоритм решения 19](#_Toc179237051)

[Тестирование 19](#_Toc179237052)

[Код программы 19](#_Toc179237053)

[Задание 3.6 20](#_Toc179237054)

[Текст задания 20](#_Toc179237055)

[Алгоритм решения 20](#_Toc179237056)

[Тестирование 20](#_Toc179237057)

[Код программы 20](#_Toc179237058)

[Задание 3.8 21](#_Toc179237059)

[Текст задания 21](#_Toc179237060)

[Алгоритм решения 21](#_Toc179237061)

[Тестирование 22](#_Toc179237062)

[Код программы 22](#_Toc179237063)

[Задание 3.10 23](#_Toc179237064)

[Текст задания 23](#_Toc179237065)

[Алгоритм решения 23](#_Toc179237066)

[Тестирование 23](#_Toc179237067)

[Код программы 24](#_Toc179237068)

[Задание 4.2 25](#_Toc179237069)

[Текст задания 25](#_Toc179237070)

[Алгоритм решения 25](#_Toc179237071)

[Тестирование 25](#_Toc179237072)

[Код программы 26](#_Toc179237073)

[Задание 4.4 27](#_Toc179237074)

[Текст задания 27](#_Toc179237075)

[Алгоритм решения 27](#_Toc179237076)

[Тестирование 27](#_Toc179237077)

[Код программы 28](#_Toc179237078)

[Задание 4.6 29](#_Toc179237079)

[Текст задания 29](#_Toc179237080)

[Алгоритм решения 29](#_Toc179237081)

[Тестирование 29](#_Toc179237082)

[Код программы 29](#_Toc179237083)

[Задание 4.8 30](#_Toc179237084)

[Текст задания 30](#_Toc179237085)

[Алгоритм решения 30](#_Toc179237086)

[Тестирование 30](#_Toc179237087)

[Код программы 30](#_Toc179237088)

[Задание 4.10 32](#_Toc179237089)

[Текст задания 32](#_Toc179237090)

[Алгоритм решения 32](#_Toc179237091)

[Тестирование 32](#_Toc179237092)

[Код программы 32](#_Toc179237093)

# Задание 1.2

## Текст задания

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:

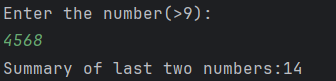
x=4568

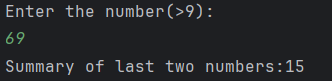
результат: 14

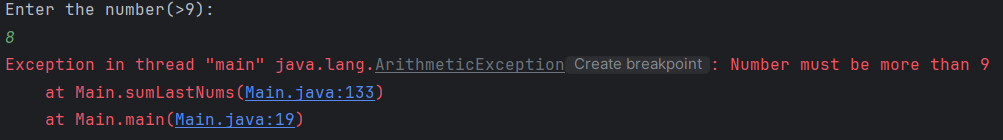
## Алгоритм решения

Складываем остаток от деления на 10 и деление нацело остатка от деления на 100 и 10, т.е. (x % 10 + (x % 100) / 10)

## Тестирование







## Код программы

<https://github.com/NTHrdd/Lab1>

# 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:

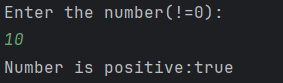
x=-5

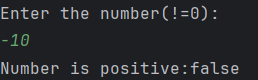
результат: false

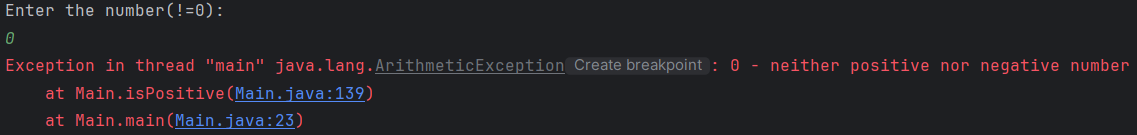
## Алгоритм решения

Возвращаем из метода булевую функцию, которая проверяет больше ли число 0.

## Тестирование







## Код программы

<https://github.com/NTHrdd/Lab1>

# 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:

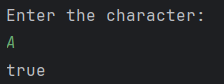
x=’q’

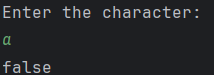
результат: false

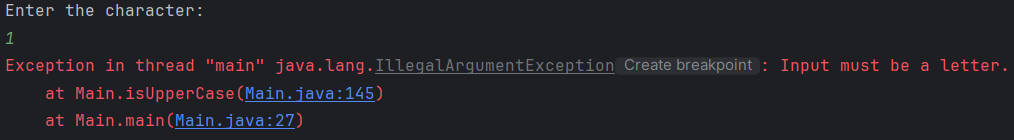
## Алгоритм решения

Возвращаем из метода булевую функцию, которая проверяет больше ли символ “A” и меньше “Z” (по ASCII).

## Тестирование







## Код программы

<https://github.com/NTHrdd/Lab1>

# 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:

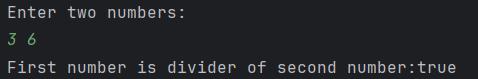
a=2 b=15

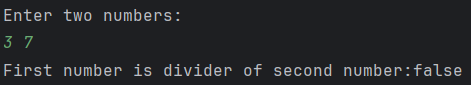
результат: false

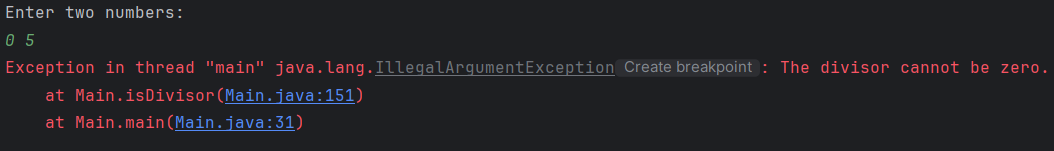
## Алгоритм решения

Возвращаем из метода булевую функцию, которая проверяет делимость b на a.

## Тестирование







## Код программы

<https://github.com/NTHrdd/Lab1>

# 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

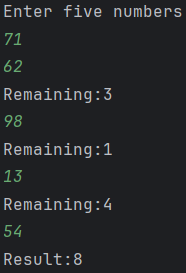
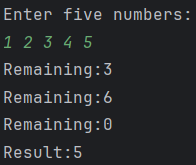
13+1 это 4

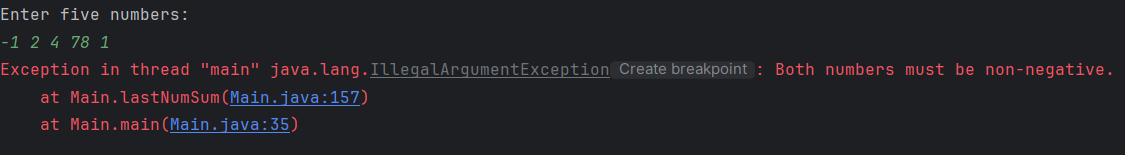
Итого 4

## Алгоритм решения

Реализуем метод, который возвращает сумму остатков от деления на 10 данных чисел. Через цикл вызываем метод 5 раз с использованием минимального количества переменных.

## Тестирование





## Код программы

<https://github.com/NTHrdd/Lab1>

# 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:

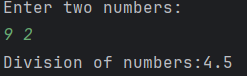
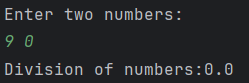
x=8 y=2

результат: 4

## Алгоритм решения

Выполняем проверку второго числа на 0. В случае успешной проверки возвращаем 0, в противном случае – деление чисел.

## Тестирование

## Код программы

<https://github.com/NTHrdd/Lab1>

# 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:

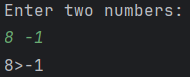
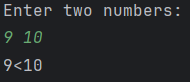
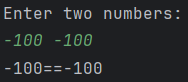
x=4 y=4

результат: “4==4”

## Алгоритм решения

Проводим 2 проверки, если x > y возвращаем строку “x > y”, если x < y возвращаем строку “x < y”, в случае, если 2 проверки вернули ложь, то возвращаем строку “x == y”.

## Тестирование

## Код программы

<https://github.com/NTHrdd/Lab1>

# 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:

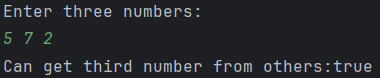
x=8 y=-1 z=4

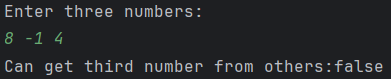
результат: false

## Алгоритм решения

Возвращаем из метода булевую функцию, которая проверяет выполнимость хотя бы одного из условий (x + y = z), (x + z = y) или (y + z = x).

## Тестирование





## Код программы

<https://github.com/NTHrdd/Lab1>

# 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:

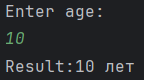
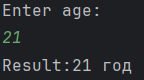
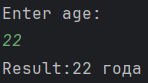
x=5

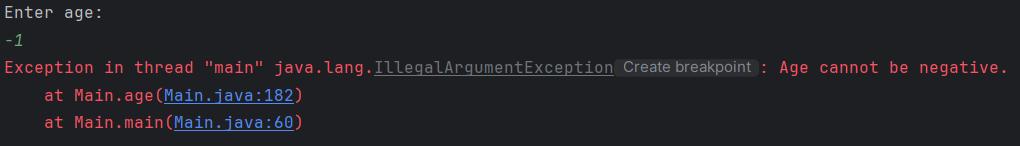
результат: “5 лет”

## Алгоритм решения

В методе реализовать 2 проверки, в условиях которых будут все проверки из описания задачи. В противном случае возвращаем строку “x лет”.

## Тестирование



## Код программы

<https://github.com/NTHrdd/Lab1>

# 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:

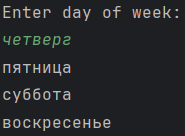
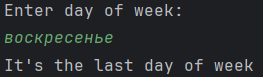
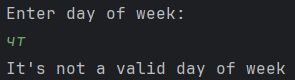
x=”чг”

результат: это не день недели

## Алгоритм решения

С помощью конструкции switch-case реализовать метод, который при попадании в определенные условия (например, x = “четверг”), будет спускаться дальше по дням неделям и выйдет в субботу, восресенье. Каждый case будет выводить последующий день недели.

## Тестирование

## Код программы

<https://github.com/NTHrdd/Lab1>

# 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:

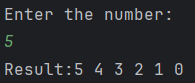
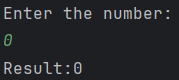
x=5

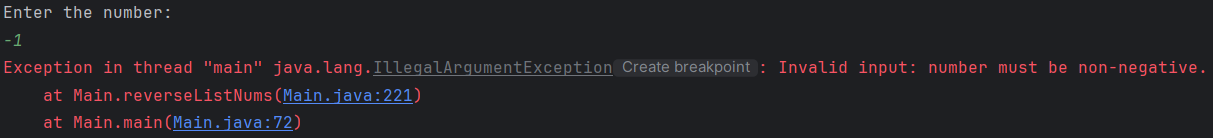
результат: “5 4 3 2 1 0”

## Алгоритм решения

Через цикл for обратным ходом формируем строку, которая содержит все числа от данного до 0.

## Тестирование



## Код программы

<https://github.com/NTHrdd/Lab1>

# 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:

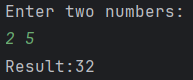
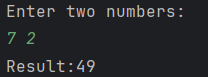
x=2 y=5

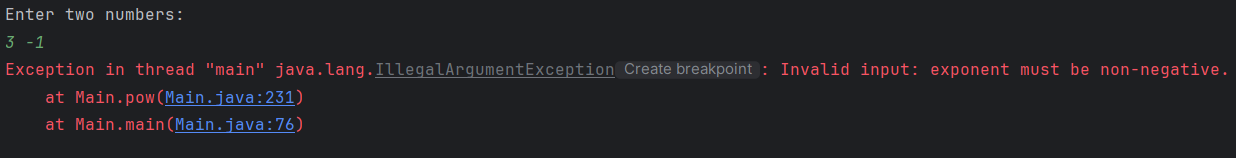
результат: 32

## Алгоритм решения

Через цикл for умножаем y раз 1 на x

## Тестирование



## Код программы

<https://github.com/NTHrdd/Lab1>

# 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:

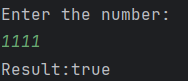
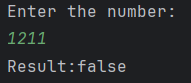
x=1211

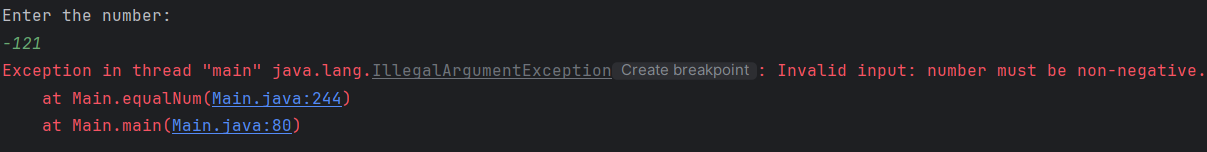
результат: false

## Алгоритм решения

Через цикл while проходим по числу и сравниваем каждую цифру с первой, т.е. каждый шаг делим число на 10 и с помощью остатка от деления проверяем новую последнюю цифру с сохраненной.

## Тестирование



## Код программы

<https://github.com/NTHrdd/Lab1>

# 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:

\*

\*\*

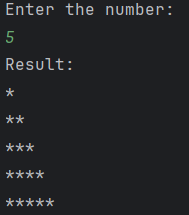
\*\*\*

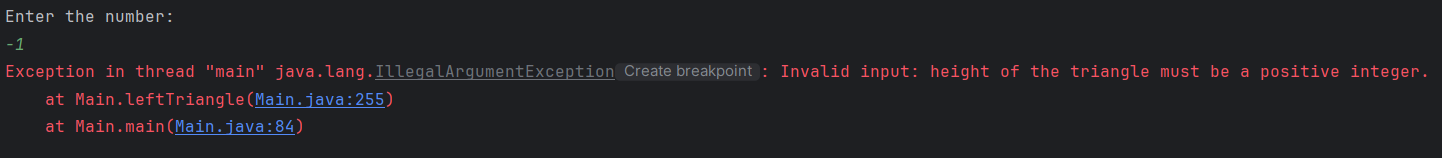
\*\*\*\*

## Алгоритм решения

Через 2 цикла for, где первый – высота треугольника, которая равна x, а второй – выводит звездочки числом равным номеру строки.

## Тестирование





## Код программы

<https://github.com/NTHrdd/Lab1>

# 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

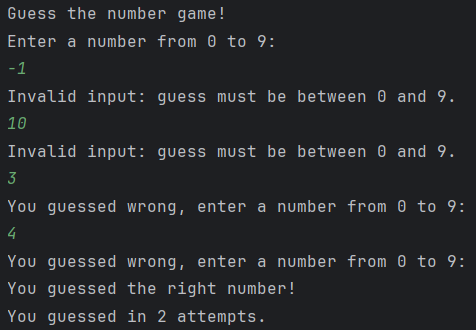
Вы угадали!

Вы отгадали число за 2 попытки

## Алгоритм решения

Генерируем случайное число. Через цикл while считываем новые числа с клавиатуры, пока не отгадаем сгенерированное. Выводим число попыток.

## Тестирование



## Код программы

<https://github.com/NTHrdd/Lab1>

# 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]

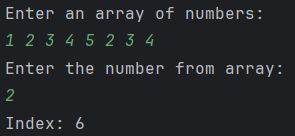
x=2

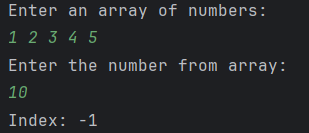
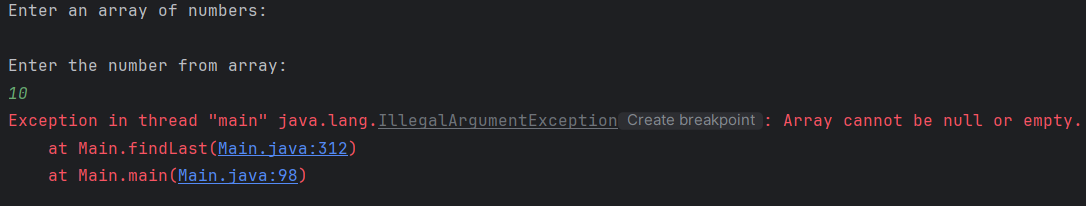
результат: 5

## Алгоритм решения

Через цикл for проходим по массиву, начиная с конца. В случае, если встречаем искомое число, выводим его индекс в массиве. В противном случае идем дальше по массиву. Если массив пройден полностью, а число отсутствует, то выводим -1.

## Тестирование



## Код программы

<https://github.com/NTHrdd/Lab1>

# 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

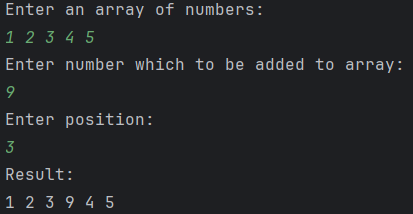
pos=3

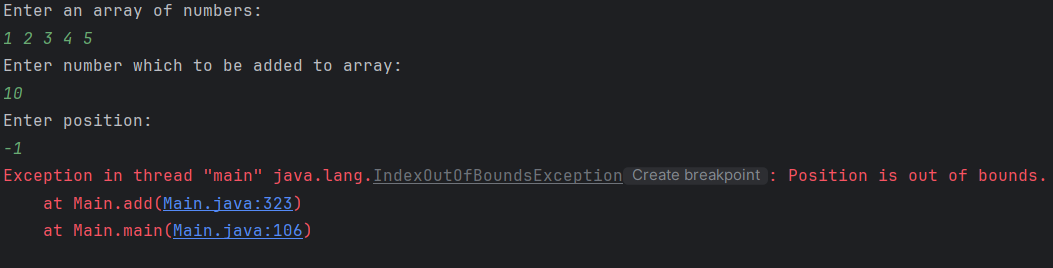
результат: [1,2,3,9,4,5]

## Алгоритм решения

Создаем новый массив, в котором первые pos членов будут совпадать с исходным, после будет вставлено новое число x, а после массив будет совпадать c исходным массивом.

## Тестирование





## Код программы

<https://github.com/NTHrdd/Lab1>

# 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:

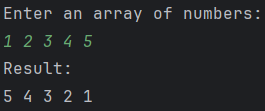
arr=[1,2,3,4,5]

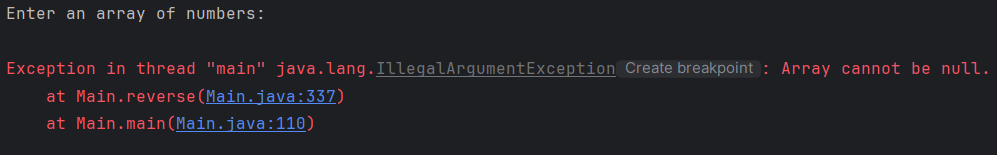
результат: arr=[5,4,3,2,1]

## Алгоритм решения

Через цикл for проходим по половине массива и меняем местами ячейки из первой половины и из второй с помощью временной переменной.

## Тестирование





## Код программы

<https://github.com/NTHrdd/Lab1>

# 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]

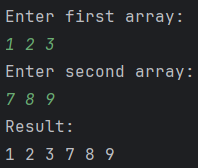
arr2=[7,8,9]

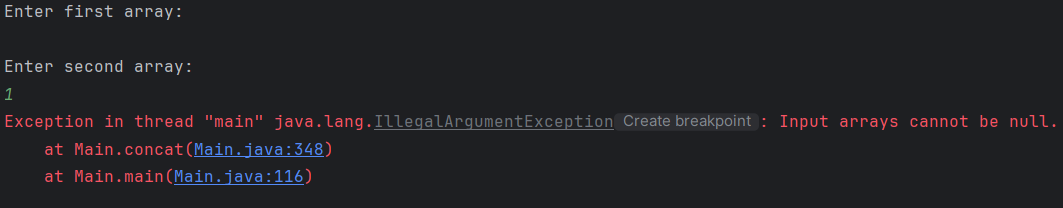
результат: [1,2,3,7,8,9]

## Алгоритм решения

Создаем новый массив, который занимает объем памяти равный исходным массивам. Через 2 цикла for проходимся по исходным массивам и копируем данные из них.

## Тестирование





## Код программы

<https://github.com/NTHrdd/Lab1>

# 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:

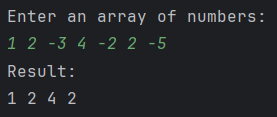
arr=[1,2,-3,4,-2,2,-5]

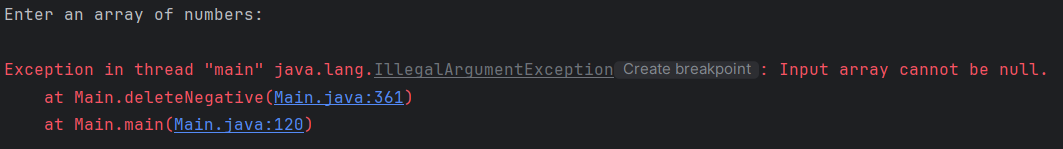
результат: [1,2,4,2]

## Алгоритм решения

Подсчитываем количество положительных чисел в массиве. Создаем новый массив длиной равной количеству положительных чисел. Копируем все положительные числа в новый массив из исходного через цикл for.

## Тестирование





## Код программы

<https://github.com/NTHrdd/Lab1>