|  |  |  |
| --- | --- | --- |
| 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 «Язык программирования Java»  Вариант 18 | | |
|  | | Работу выполнил  студент группы ФИТ-2-2024 1 курса  Маткин М.А.  «03» октября 2024 г. |
| Работу проверил  Ракина В.Д.  «\_\_\_» \_\_\_\_\_\_\_\_\_ 2024 г. |
| Пермь 2024 | | |

СОДЕРЖАНИЕ

[Задание 1.2 6](#_Toc178897459)

[Текст задания 6](#_Toc178897460)

[Алгоритм решения 6](#_Toc178897461)

[Тестирование 6](#_Toc178897462)

[Код программы 7](#_Toc178897463)

[Задание 1.4 8](#_Toc178897464)

[Текст задания 8](#_Toc178897465)

[Алгоритм решения 8](#_Toc178897466)

[Тестирование 8](#_Toc178897467)

[Код программы 9](#_Toc178897468)

[Задание 1.6 10](#_Toc178897469)

[Текст задания 10](#_Toc178897470)

[Алгоритм решения 10](#_Toc178897471)

[Тестирование 11](#_Toc178897472)

[Код программы 11](#_Toc178897473)

[Задание 1.8 12](#_Toc178897474)

[Текст задания 12](#_Toc178897475)

[Алгоритм решения 12](#_Toc178897476)

[Тестирование 12](#_Toc178897477)

[Код программы 13](#_Toc178897478)

[Задание 1.10 14](#_Toc178897479)

[Текст задания 14](#_Toc178897480)

[Алгоритм решения 14](#_Toc178897481)

[Тестирование 15](#_Toc178897482)

[Код программы 15](#_Toc178897483)

[Задание 2.2 16](#_Toc178897484)

[Текст задания 16](#_Toc178897485)

[Алгоритм решения 16](#_Toc178897486)

[Тестирование 17](#_Toc178897487)

[Код программы 17](#_Toc178897488)

[Задание 2.4 18](#_Toc178897489)

[Текст задания 18](#_Toc178897490)

[Алгоритм решения 18](#_Toc178897491)

[Тестирование 19](#_Toc178897492)

[Код программы 19](#_Toc178897493)

[Задание 2.6 20](#_Toc178897494)

[Текст задания 20](#_Toc178897495)

[Алгоритм решения 20](#_Toc178897496)

[Тестирование 21](#_Toc178897497)

[Код программы 21](#_Toc178897498)

[Задание 2.8 22](#_Toc178897499)

[Текст задания 22](#_Toc178897500)

[Алгоритм решения 22](#_Toc178897501)

[Тестирование 23](#_Toc178897502)

[Код программы 23](#_Toc178897503)

[Задание 2.10 24](#_Toc178897504)

[Текст задания 24](#_Toc178897505)

[Алгоритм решения 24](#_Toc178897506)

[Тестирование 25](#_Toc178897507)

[Код программы 25](#_Toc178897508)

[Задание 3.2 26](#_Toc178897509)

[Текст задания 26](#_Toc178897510)

[Алгоритм решения 26](#_Toc178897511)

[Тестирование 26](#_Toc178897512)

[Код программы 26](#_Toc178897513)

[Задание 3.4 27](#_Toc178897514)

[Текст задания 27](#_Toc178897515)

[Алгоритм решения 27](#_Toc178897516)

[Тестирование 28](#_Toc178897517)

[Код программы 28](#_Toc178897518)

[Задание 3.6 29](#_Toc178897519)

[Текст задания 29](#_Toc178897520)

[Алгоритм решения 29](#_Toc178897521)

[Тестирование 30](#_Toc178897522)

[Код программы 30](#_Toc178897523)

[Задание 3.8 31](#_Toc178897524)

[Текст задания 31](#_Toc178897525)

[Алгоритм решения 31](#_Toc178897526)

[Тестирование 32](#_Toc178897527)

[Код программы 32](#_Toc178897528)

[Задание 3.10 33](#_Toc178897529)

[Текст задания 33](#_Toc178897530)

[Алгоритм решения 33](#_Toc178897531)

[Тестирование 33](#_Toc178897532)

[Код программы 34](#_Toc178897533)

[Задание 4.2 35](#_Toc178897534)

[Текст задания 35](#_Toc178897535)

[Алгоритм решения 35](#_Toc178897536)

[Тестирование 35](#_Toc178897537)

[Код программы 36](#_Toc178897538)

[Задание 4.4 37](#_Toc178897539)

[Текст задания 37](#_Toc178897540)

[Алгоритм решения 37](#_Toc178897541)

[Тестирование 37](#_Toc178897542)

[Код программы 38](#_Toc178897543)

[Задание 4.6 39](#_Toc178897544)

[Текст задания 39](#_Toc178897545)

[Алгоритм решения 39](#_Toc178897546)

[Тестирование 39](#_Toc178897547)

[Код программы 39](#_Toc178897548)

[Задание 4.8 41](#_Toc178897549)

[Текст задания 41](#_Toc178897550)

[Алгоритм решения 41](#_Toc178897551)

[Тестирование 42](#_Toc178897552)

[Код программы 42](#_Toc178897553)

[Задание 4.10 43](#_Toc178897554)

[Текст задания 43](#_Toc178897555)

[Алгоритм решения 43](#_Toc178897556)

[Тестирование 43](#_Toc178897557)

[Код программы 44](#_Toc178897558)

# 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:

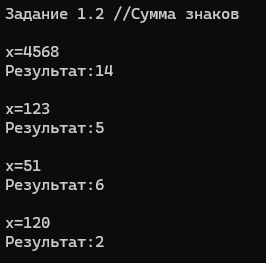
x=4568

результат: 14

## Алгоритм решения

Возвращаем x/10%10 + x%10.

## Тестирование



## Код программы

https://github.com/NPNN00/javaPSU/blob/dev/labs/src/Lab1.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

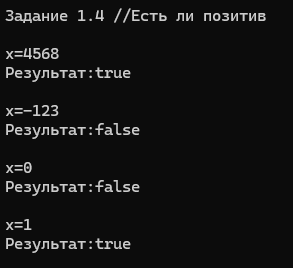
Пример 2: x=-5

результат: false

## Алгоритм решения

Возвращаем x>0.

## Тестирование



## Код программы

https://github.com/NPNN00/javaPSU/blob/dev/labs/src/Lab1.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:

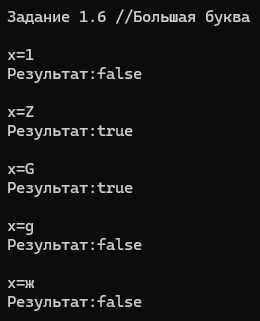
x=’q’

результат: false

## Алгоритм решения

Возвращаем x>=’A’ && x<=’Z’.

## Тестирование



## Код программы

https://github.com/NPNN00/javaPSU/blob/dev/labs/src/Lab1.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

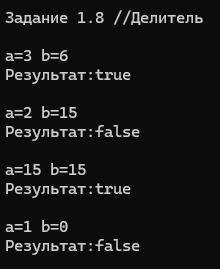
результат: false

## Алгоритм решения

Возвращаем a!=0 && b!=0 && (a%b == 0 || b%a == 0).

Дополнительно добавили проверку a и b на 0.

## Тестирование



## Код программы

https://github.com/NPNN00/javaPSU/blob/dev/labs/src/Lab1.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

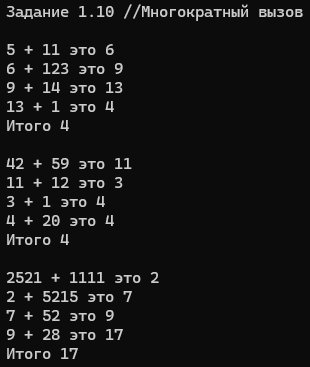
13+1 это 4

Итого 4

## Алгоритм решения

Используем две переменные – для запоминания старого результата и для ввода нового числа. Вызываем метод суммы цифр в цикле от этих аргументов.

## Тестирование



## Код программы

https://github.com/NPNN00/javaPSU/blob/dev/labs/src/Lab1.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:

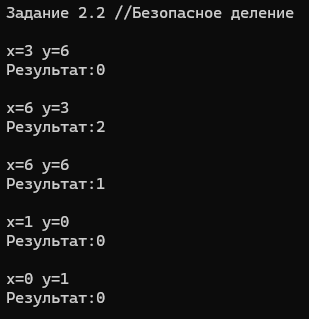
x=8 y=2

результат: 4

## Алгоритм решения

Добавляем проверку делителя на 0. Если он ему не равен, то возвращаем результат деления.

## Тестирование



## Код программы

https://github.com/NPNN00/javaPSU/blob/dev/labs/src/Lab1.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:

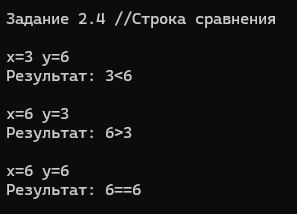
x=4 y=4

результат: “4==4”

## Алгоритм решения

Используем конструкцию switch-case и результат сравнения метода Integer.compare(x,y), который возвращает -1 при x>y, 0 при x==y и 1 при x<1 .

## Тестирование



## Код программы

https://github.com/NPNN00/javaPSU/blob/dev/labs/src/Lab1.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:

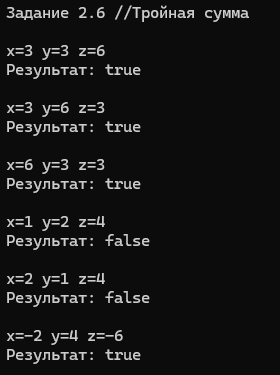
x=8 y=-1 z=4

результат: false

## Алгоритм решения

Возвращаем x+y==z || x+z==y ||y+z==x.

## Тестирование



## Код программы

https://github.com/NPNN00/javaPSU/blob/dev/labs/src/Lab1.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:

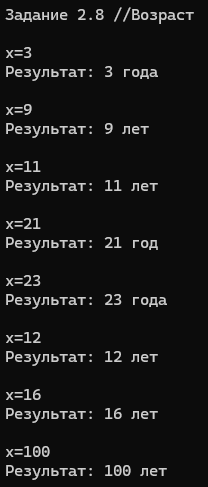
x=44

результат: “44 года”

## Алгоритм решения

Добавляем в метод условные конструкции чётко по заданию и возвращаем результат.

## Тестирование



## Код программы

https://github.com/NPNN00/javaPSU/blob/dev/labs/src/Lab1.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”

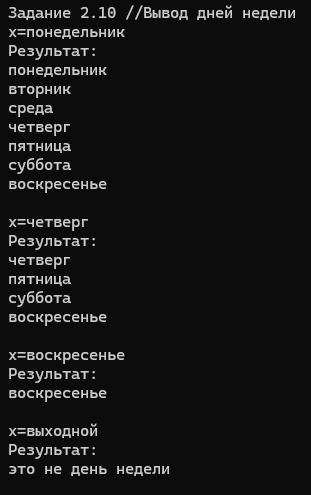
результат:

это не день недели

## Алгоритм решения

Используем конструкцию switch-case(x), где x – день недели. В каждом case по дню недели выводим результат и проваливаемся вниз (необходимо не использовать break с понедельника до субботы включительно). Остальные случаи обрабатываем с помощью default.

## Тестирование



## Код программы

https://github.com/NPNN00/javaPSU/blob/dev/labs/src/Lab1.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:

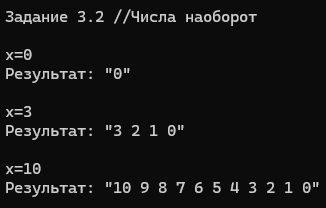
x=5

результат: “5 4 3 2 1 0”

## Алгоритм решения

Проходим в цикле от x до нуля и строим результат в виде строки. Метод trim() удаляет пробелы в начале и в конце строки.

## Тестирование



## Код программы

https://github.com/NPNN00/javaPSU/blob/dev/labs/src/Lab1.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:

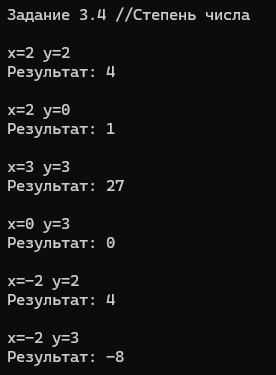
x=2 y=5

результат: 32

## Алгоритм решения

Инициализируем результат = 1, затем проходим в цикле от 0 до степени y, умножая результат на число x.

## Тестирование



## Код программы

https://github.com/NPNN00/javaPSU/blob/dev/labs/src/Lab1.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:

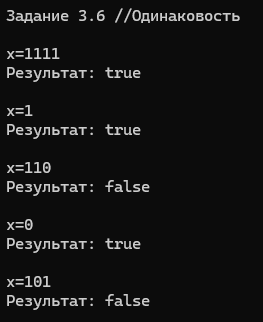
x=1211

результат: false

## Алгоритм решения

Сравниваем в цикле последние две соседние цифры в числе. Если не равны – возвращаем false. Если все оказались равны, выходим из цикла и возвращаем true.

## Тестирование



## Код программы

https://github.com/NPNN00/javaPSU/blob/dev/labs/src/Lab1.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:

\*

\*\*

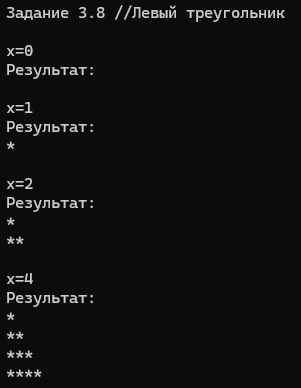
\*\*\*

\*\*\*\*

## Алгоритм решения

Используем вложенный цикл.

## Тестирование



## Код программы

https://github.com/NPNN00/javaPSU/blob/dev/labs/src/Lab1.java

# Задание 3.10

## Текст задания

Угайдака.

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!

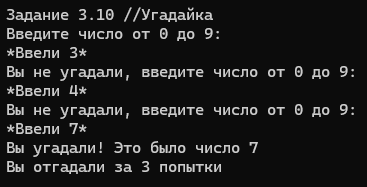
Вы отгадали число за 2 попытки

## Алгоритм решения

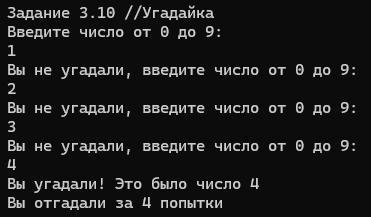
Инициализируем количество попыток = 0 и случайное целое число от 0 до 9. Затем в бесконечном цикле запрашиваем результат. Если угадали – выходим из цикла. В каждой итерации цикла инкрементируем количество попыток на 1.

## Тестирование

Автоматический ввод:



Ввод с клавиатуры:



## Код программы

https://github.com/NPNN00/javaPSU/blob/dev/labs/src/Lab1.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]

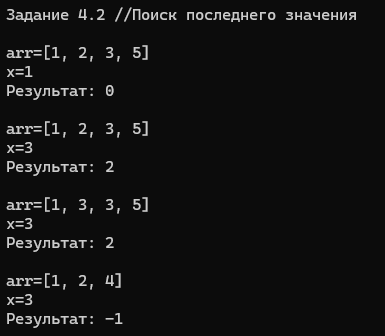
x=2

результат: 5

## Алгоритм решения

Инициализируем результат = -1. Проходим по массиву и сохраняем в результат индекс элемента, если он равен x.

## Тестирование



## Код программы

https://github.com/NPNN00/javaPSU/blob/dev/labs/src/Lab1.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

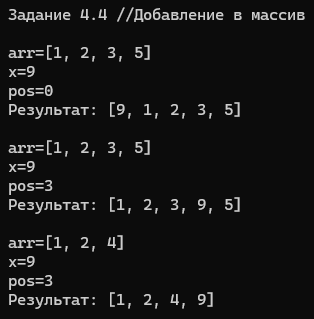
pos=3

результат: [1,2,3,9,4,5]

## Алгоритм решения

Создаём массив размера arr.length+1, затем в цикле заполняем числа до нужной позиции, отдельно вставляем новый элемент, потом заполняем в цикле оставшиеся числа.

## Тестирование



## Код программы

https://github.com/NPNN00/javaPSU/blob/dev/labs/src/Lab1.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:

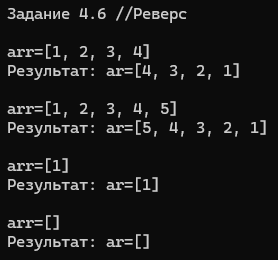
arr=[1,2,3,4,5]

результат: arr=[5,4,3,2,1]

## Алгоритм решения

Идём с начала до середины массива, меняя значения на зеркальный элемент от середины.

## Тестирование



## Код программы

https://github.com/NPNN00/javaPSU/blob/dev/labs/src/Lab1.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]

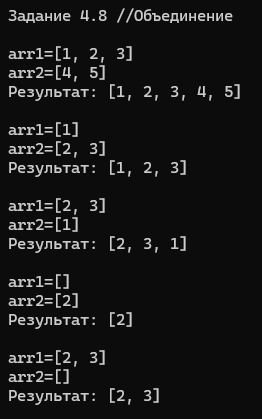
arr2=[7,8,9]

результат: [1,2,3,7,8,9]

## Алгоритм решения

Создаем массив, у которого размерность равна сумме размерностей двух массивов и заполняем его при помощи двух циклов.

## Тестирование



## Код программы

https://github.com/NPNN00/javaPSU/blob/dev/labs/src/Lab1.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:

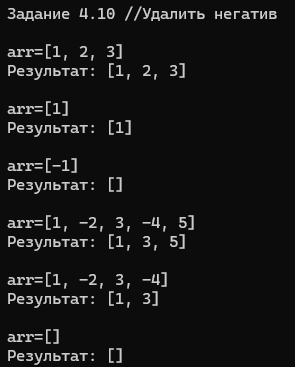
arr=[1,2,-3,4,-2,2,-5]

результат: [1,2,4,2]

## Алгоритм решения

Изначально считаем количество НЕ негативных элементов в массиве, затем создаём новый массив с размерностью, равной этому числу НЕ негативных элементов. Далее в цикле добавляем нужные элементы, а негативные пропускаем.

## Тестирование



## Код программы

https://github.com/NPNN00/javaPSU/blob/dev/labs/src/Lab1.java