|  |  |  |
| --- | --- | --- |
| 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 Java»  Нечетный вариант | | |
|  | | Работу выполнил  студент группы ДВБ7-ПР3-2023 НБ  2 курса  Пшеничникова А.И.  «\_\_\_» \_\_\_\_\_\_\_\_2024 г. |
| Работу проверила  Ракина В.Д.  «\_\_\_» \_\_\_\_\_\_\_\_\_ 2024 г. |
| Пермь 2024 | | |

СОДЕРЖАНИЕ

[Задание 1.1 5](#_Toc179835292)

[Текст задания 5](#_Toc179835293)

[Алгоритм решения 5](#_Toc179835294)

[Тестирование 5](#_Toc179835295)

[Задание 1.3 5](#_Toc179835296)

[Текст задания 5](#_Toc179835297)

[Алгоритм решения 5](#_Toc179835298)

[Тестирование 5](#_Toc179835299)

[Задание 1.5 5](#_Toc179835300)

[Текст задания 5](#_Toc179835301)

[Алгоритм решения 5](#_Toc179835302)

[Тестирование 5](#_Toc179835303)

[Задание 1.7 5](#_Toc179835304)

[Текст задания 5](#_Toc179835305)

[Алгоритм решения 5](#_Toc179835306)

[Тестирование 5](#_Toc179835307)

[Задание 1.9 5](#_Toc179835308)

[Текст задания 5](#_Toc179835309)

[Алгоритм решения 5](#_Toc179835310)

[Тестирование 5](#_Toc179835311)

[Задание 2.1 5](#_Toc179835312)

[Текст задания 5](#_Toc179835313)

[Алгоритм решения 5](#_Toc179835314)

[Тестирование 5](#_Toc179835315)

[Задание 2.3 5](#_Toc179835316)

[Текст задания 5](#_Toc179835317)

[Алгоритм решения 5](#_Toc179835318)

[Тестирование 5](#_Toc179835319)

[Задание 2.5 5](#_Toc179835320)

[Текст задания 5](#_Toc179835321)

[Алгоритм решения 5](#_Toc179835322)

[Тестирование 5](#_Toc179835323)

[Задание 2.7 5](#_Toc179835324)

[Текст задания 5](#_Toc179835325)

[Алгоритм решения 5](#_Toc179835326)

[Тестирование 5](#_Toc179835327)

[Задание 2.9 5](#_Toc179835328)

[Текст задания 5](#_Toc179835329)

[Алгоритм решения 5](#_Toc179835330)

[Тестирование 5](#_Toc179835331)

[Задание 3.1 5](#_Toc179835332)

[Текст задания 5](#_Toc179835333)

[Алгоритм решения 5](#_Toc179835334)

[Тестирование 5](#_Toc179835335)

[Задание 3.3 5](#_Toc179835336)

[Текст задания 5](#_Toc179835337)

[Алгоритм решения 5](#_Toc179835338)

[Тестирование 5](#_Toc179835339)

[Задание 3.5 6](#_Toc179835340)

[Текст задания 6](#_Toc179835341)

[Алгоритм решения 6](#_Toc179835342)

[Тестирование 6](#_Toc179835343)

[Задание 3.7 6](#_Toc179835344)

[Текст задания 6](#_Toc179835345)

[Алгоритм решения 6](#_Toc179835346)

[Тестирование 6](#_Toc179835347)

[Задание 3.9 6](#_Toc179835348)

[Текст задания 6](#_Toc179835349)

[Алгоритм решения 6](#_Toc179835350)

[Тестирование 6](#_Toc179835351)

[Задание 4.1 6](#_Toc179835352)

[Текст задания 6](#_Toc179835353)

[Алгоритм решения 6](#_Toc179835354)

[Тестирование 6](#_Toc179835355)

[Задание 4.3 6](#_Toc179835356)

[Текст задания 6](#_Toc179835357)

[Алгоритм решения 6](#_Toc179835358)

[Тестирование 6](#_Toc179835359)

[Задание 4.5 6](#_Toc179835360)

[Текст задания 6](#_Toc179835361)

[Алгоритм решения 6](#_Toc179835362)

[Тестирование 6](#_Toc179835363)

[Задание 4.7 6](#_Toc179835364)

[Текст задания 6](#_Toc179835365)

[Алгоритм решения 6](#_Toc179835366)

[Тестирование 6](#_Toc179835367)

[Задание 4.9 6](#_Toc179835368)

[Текст задания 6](#_Toc179835369)

[Алгоритм решения 6](#_Toc179835370)

[Тестирование 6](#_Toc179835371)

[Код программы 6](#_Toc179835372)

# Задание 1.1

## Текст задания

Дробная часть.

Дана сигнатура метода: public double fraction (double x); Необходимо реализовать метод таким образом, чтобы он возвращал только дробную часть числа х. Подсказка: вещественное число может быть преобразовано к целому путем отбрасывания дробной части.

Пример: x=5,25; результат: 0,25

## Алгоритм решения

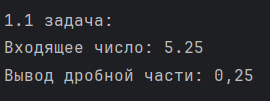
1. Преобразуем входное число x в целую и дробную части:

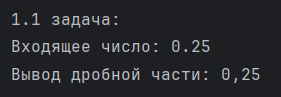
Целая часть: (int)x

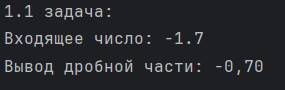
Дробная часть: x - (int)x

2. Возвращаем только дробную часть.

## Тестирование







# Задание 1.3

## Текст задания

Букву в число.

Дана сигнатура метода: public int charToNum (char x); Метод принимает символ х, который представляет собой один из “0 1 2 3 4 5 6 7 8 9”. Необходимо реализовать метод таким образом, чтобы он преобразовывал символ в соответствующее число. Подсказка: код символа ‘0’ — это число 48.

Пример: x=’3’; результат: 3

## Алгоритм решения

1. Проверка входного символа:

Убедимся, что символ является цифрой от '0' до '9'.

Если нет, выбрасываем IllegalArgumentException.

2. Преобразование символа в число:

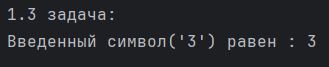
Используем операцию вычитания кода символа из кода нуля ('0').

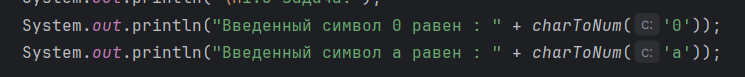
Это дает нам прямое числовое значение символа.

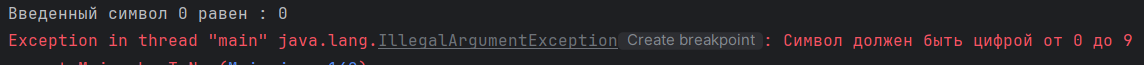
3. Возврат результата:

Вернем полученное числовое значение.

## Тестирование







# Задание 1.5

## Текст задания

Двузначное.

Дана сигнатура метода: public bool is2Digits (int x); Необходимо реализовать метод таким образом, чтобы он принимал число x и возвращал true, если оно двузначное.

Пример 1: x=32; результат: true

Пример 2: x=516; результат: false

## Алгоритм решения

1. Проверка условия:

Убедимся, что число x больше или равно 10 и меньше или равно 99.

Если число находится в этом диапазоне, оно двузначное.

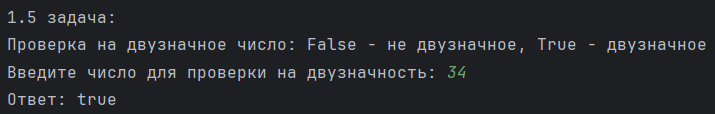
2. Возврат результата:

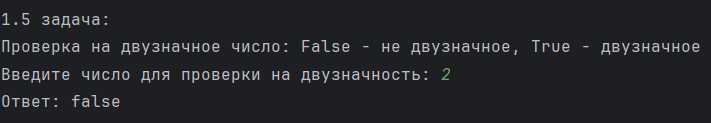
Вернем true, если условие выполняется.

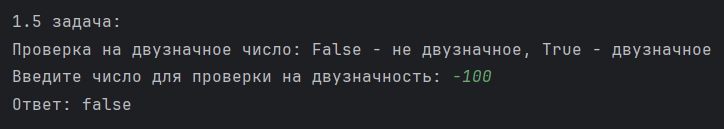
Иначе вернем false.

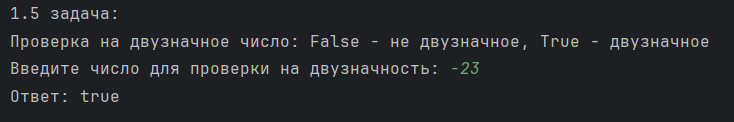
## Тестирование

Скриншоты результата работы программы.









# Задание 1.7

## Текст задания

Диапазон.

Дана сигнатура метода: public bool isInRange (int a, int b, int num); Метод принимает левую и правую границу (a и b) некоторого числового диапазона. Необходимо реализовать метод таким образом, чтобы он возвращал true, если num входит в указанный диапазон (включая границы). Обратите внимание, что отношение a и b заранее неизвестно (неясно кто из них больше, а кто меньше)

Пример 1: a=5 b=1 num=3 результат: true

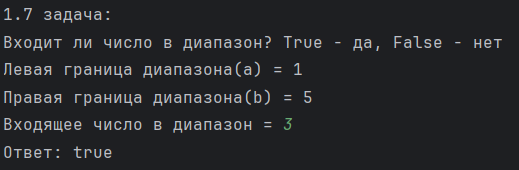
Пример 2: a=2 b=15 num=33 результат: false

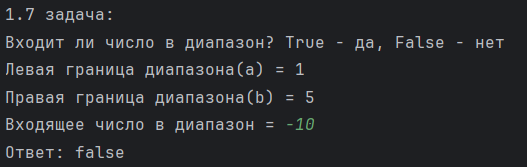
## Алгоритм решения

1. Сначала мы должны определить, какая из границ (a или b) является меньшей, а какая большей.

2. Затем мы используем условные операторы для проверки, находится ли num в диапазоне.

## Тестирование





# Задание 1.9

## Текст задания

Равенство.

Дана сигнатура метода: public bool isEqual(int a, int b, int c); Необходимо реализовать метод таким образом, чтобы он возвращал true, если все три полученных методом числа равны

Пример 1: a=3 b=3 с=3 результат: true

Пример 2: a=2 b=15 с=2 результат: false

## Алгоритм решения

Алгоритм решения задачи.

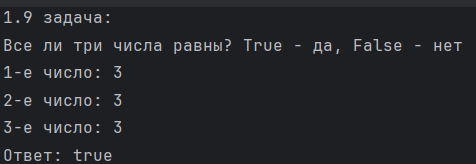
1. Создать метод isEqual с тремя параметрами типа int.

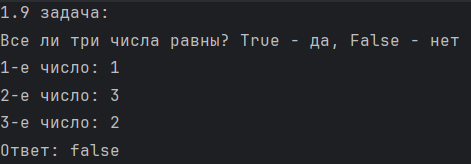
2. Внутри метода использовать оператор сравнения для проверки равенства трех чисел.

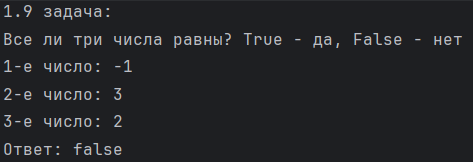
3. Возвращать true, если все числа равны, иначе false.

## Тестирование

Скриншоты результата работы программы.







# Задание 2.1

## Текст задания

Модуль числа.

Дана сигнатура метода: public int abs (int x); Необходимо реализовать метод таким образом, чтобы он возвращал модуль числа х (если оно было положительным, то таким и остается, если он было отрицательным – то необходимо вернуть его без знака минус).

Пример 1: x=5 результат: 5

Пример 2: x=-3 результат: 3

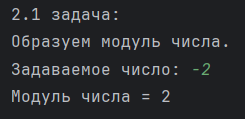
## Алгоритм решения

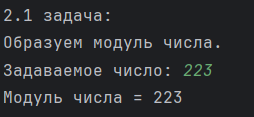
1. Создаем метод abs(int x) с параметром типа int.

2. Если число неотрицательное (x >= 0), мы просто возвращаем его без изменений.

3. Если число отрицательное (x < 0), мы умножаем его на -1, что меняет знак с минус на плюс.

## Тестирование





# Задание 2.3

## Текст задания

Тридцать пять.

Дана сигнатура метода: public bool is35 (int x); Необходимо реализовать метод таким образом, чтобы он возвращал true, если число x делится нацело на 3 или 5. При этом, если оно делится и на 3, и на 5, то вернуть надо false. Подсказка: оператор % позволяет получить остаток от деления.

Пример 1: x=5 результат: true

Пример 2: x=8 результат: false

Пример 3: x=15 результат: false

## Алгоритм решения

1. Проверяем, делятся ли число x на 3:

Если да, то проверяем, делятся ли оно также на 5

Если нет, то возвращаем true

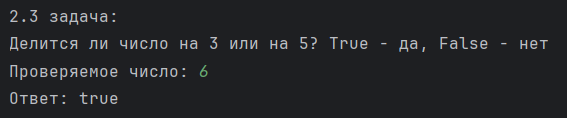
2. Если число не делятся на 3, то:

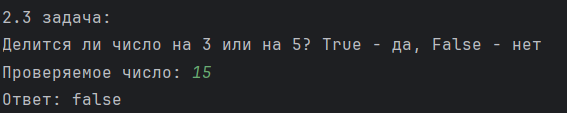
Проверяем, делятся ли оно на 5

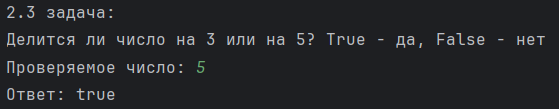
Если да, то возвращаем false

Если нет, то возвращаем true

## Тестирование







# Задание 2.5

## Текст задания

Тройной максимум.

Дана сигнатура метода: public int max3 (int x, int y, int z); Необходимо реализовать метод таким образом, чтобы он возвращал максимальное из трех полученных методом чисел. Подсказка: идеальное решение включает всего две инструкции if и не содержит вложенных if.

Пример 1: x=5 y=7 z=7 результат: 7

Пример 2: x=8 y=-1 z=4 результат: 8

## Алгоритм решения

Алгоритм решения задачи.

1. Сначала проверяем, больше ли x, чем y и z.

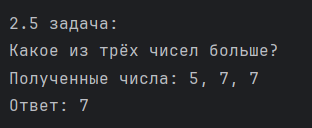
2. Если да, возвращаем x как максимальное число.

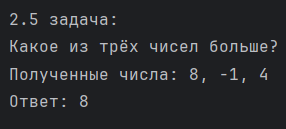
3. Если нет, то проверяем, больше ли y, чем x и z.

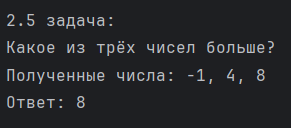
4. Если y больше, возвращаем y как максимальное число.

5. В противном случае, возвращаем z как максимальное число.

## Тестирование







# Задание 2.7

## Текст задания

Двойная сумма.

Дана сигнатура метода: public int sum2 (int x, int y); Необходимо реализовать метод таким образом, чтобы он возвращал сумму чисел x и y. Однако, если сумма попадает в диапазон от 10 до 19, то надо вернуть число 20.

Пример 1: x=5 y=7 результат: 20

Пример 2: x=8 y=-1 результат: 7

## Алгоритм решения

Алгоритм решения задачи.

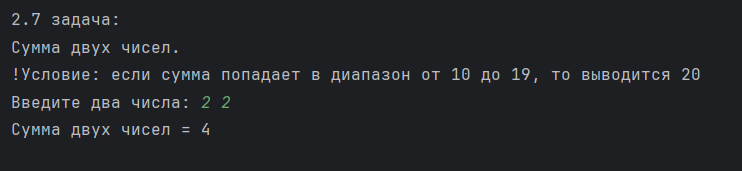
1. Создаем метод sum2 с сигнатурой public int sum2(int x, int y).

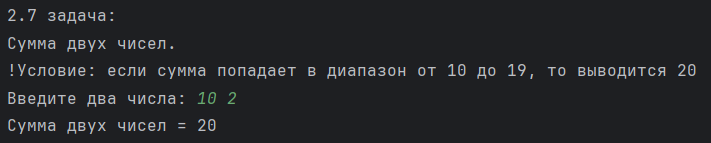
2. Вычисляем сумму чисел x и y.

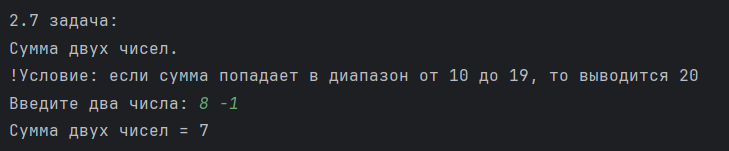
3. Проверяем, попала ли сумма в диапазон от 10 до 19 включительно.

Если сумма в указанном диапазоне – возвращаем 20, иначе возвращаем саму сумму.

## Тестирование







# Задание 2.9

## Текст задания

День недели.

Дана сигнатура метода: public String day (int x); Метод принимает число x, обозначающее день недели. Необходимо реализовать метод таким образом, чтобы он возвращал строку, которая будет обозначать текущий день недели, где 1- это понедельник, а 7 – воскресенье. Если число не от 1 до 7 то верните текст “это не день недели”. Вместо if в данной задаче используйте switch.

Пример: x=5 результат: “пятница”

## Алгоритм решения

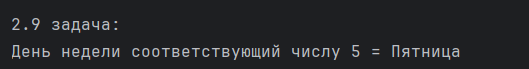
1. Создаем switch блок с случаями от 1 до 7.

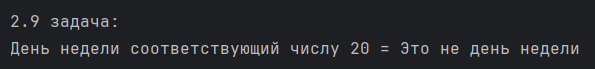
2. Для каждого случая возвращаем соответствующую строку дня недели.

3. Добавляем default case для вывода "Это не день недели".

## Тестирование

Скриншоты результата работы программы.





# Задание 3.1

## Текст задания

Числа подряд.

Дана сигнатура метода: public String listNums (int x); Необходимо реализовать метод таким образом, чтобы он возвращал строку, в которой будут записаны все числа от 0 до x (включительно).

Пример: x=5 результат: “0 1 2 3 4 5”

## Алгоритм решения

1. Создаем метод listNums(int x) с возвращаемым типом String.

2. Используем StringBuilder для эффективного построения строки чисел.

3. Итерируем от 0 до x включительно.

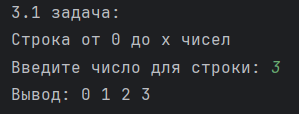
4. Для каждого числа:

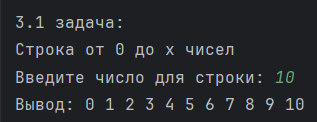
Если это не первое число, добавляем пробел перед ним.

Добавляем само число к StringBuilder.

5. Возвращаем строку из StringBuilder как результат.

## Тестирование





# Задание 3.3

## Текст задания

Четные числа.

Дана сигнатура метода: public String chet (int x); Необходимо реализовать метод таким образом, чтобы он возвращал строку, в которой будут записаны все четные числа от 0 до x (включительно). Подсказа для обеспечения качества кода: инструкцию if использовать не следует.

Пример: x=9 результат: “0 2 4 6 8”

## Алгоритм решения

Алгоритм решения задачи.

1. Создаем новый StringBuilder result.

2. Начинаем цикл от 0 до x с шагом 2.

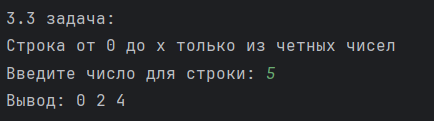
3. Для каждого четного числа:

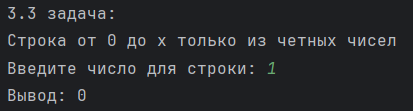
Проверяем, пустая ли текущая строка. Если нет, добавляем пробел.

Добавляем число к текущей строке.

4. После цикла возвращаем содержимое StringBuilder как строку.

## Тестирование





# Задание 3.5

## Текст задания

Длина числа.

Дана сигнатура метода: public int numLen (long x); Необходимо реализовать метод таким образом, чтобы он возвращал количество знаков в числе x. Подсказка: Int у=123/10; // у будет иметь значение 12

Пример: x=12567 результат: 5

## Алгоритм решения

Алгоритм решения задачи.

1. Сначала мы проверяем, не нулевое ли число. Если да, то оно имеет одну цифру.

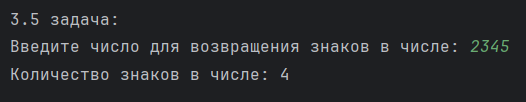
2. Мы преобразуем абсолютное значение числа в строку, чтобы избежать проблем с отрицательными числами.

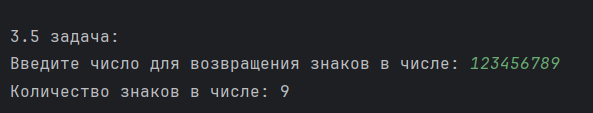
3. Затем мы перебираем каждый символ в строке числа.

4. Мы увеличиваем счетчик только для символов, которые являются цифрами (не пробелы).

## Тестирование

Скриншоты результата работы программы.





# Задание 3.7

## Текст задания

Квадрат.

Дана сигнатура метода: public void square (int x); Необходимо реализовать метод таким образом, чтобы он выводил на экран квадрат из символов ‘\*’ размером х, у которого х символов в ряд и х символов в высоту.

## Алгоритм решения

1. Создаем метод square() без параметров.

2. Внутри метода создаем объект Scanner для ввода данных пользователем.

3. Выводим запрос на ввод числа x.

4. Прочитаем введенное число x.

5. Используем два цикла for для печати квадрата:

Внешний цикл для строк (x раз).

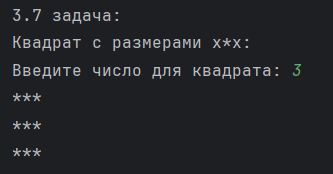
Внутренний цикл для символов в строке (x раз).

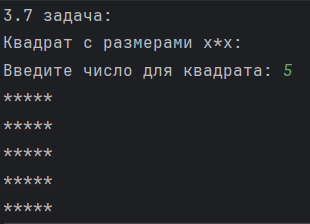
6. Внутри внутреннего цикла печатаем символ «\*».

7. После каждого заполненного ряда перенос строки.

## Тестирование

Скриншоты результата работы программы.





# Задание 3.9

## Текст задания

Правый треугольник.

Дана сигнатура метода: public void rightTriangle (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, при этом треугольник выровнен по правому краю. Подсказка: перед символами ‘\*’ следует выводить необходимое количество пробелов.

## Алгоритм решения

1. Создаем параметр x, который будет представлять высоту треугольника.

2. Используем внешний цикл от 1 до x для генерации строк треугольника.

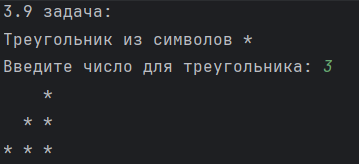
3. Для каждой строки:

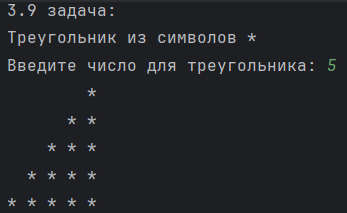
Внутренний цикл для вывода пробелов слева от звездочек.

Второй внутренний цикл для вывода звездочек.

4. Выводим новую строку после каждой полной строки треугольника.

## Тестирование





# Задание 4.1

## Текст задания

Поиск первого значения.

Дана сигнатура метода: public int findFirst (int[] arr, int x); Необходимо реализовать метод таким образом, чтобы он возвращал индекс первого 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: 1

## Алгоритм решения

1. Инициализируем переменную index равной -1.

2. Перебираем массив от начала до конца.

3. При каждой итерации проверяем, равен ли текущий элемент x.

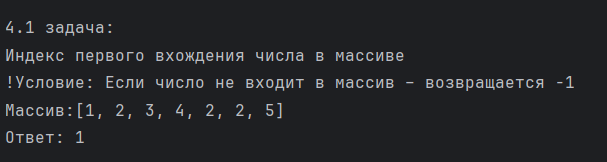
4. Если элемент равен x, возвращаем текущий индекс.

5. Если элемент не равен x, продолжаем перебор.

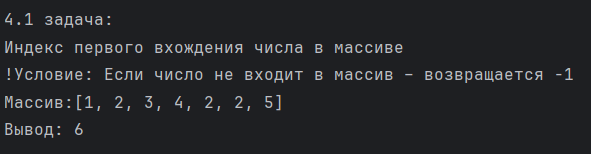
6. После завершения перебора, если index все еще равен -1, выводим -1.

## Тестирование

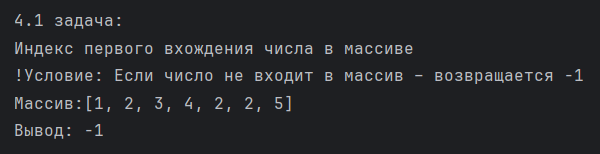
X = 2



X = 5



X = 7



# Задание 4.3

## Текст задания

Поиск максимального.

Дана сигнатура метода: public int maxAbs (int[] arr); Необходимо реализовать метод таким образом, чтобы он возвращал наибольшее по модулю (то есть без учета знака) значение массива arr.

Пример: arr=[1,-2,-7,4,2,2,5] результат: -7

## Алгоритм решения

1. Инициализируем переменную maxAbs с нулевым значением.

2. Проходим по всем элементам массива arr:

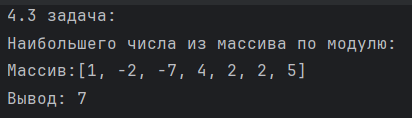
Если текущий элемент меньше нуля, преобразуем его в положительное число.

Сравниваем текущее значение с maxAbs. Если оно больше, обновляем maxAbs.

3. После прохода по всему массиву возвращаем maxAbs.

## Тестирование

Скриншоты результата работы программы.



# Задание 4.5

## Текст задания

Добавление массива в массив.

Дана сигнатура метода: public int[] add (int[] arr, int[] ins, int pos); Необходимо реализовать метод таким образом, чтобы он возвращал новый массив, который будет содержать все элементы массива arr, однако в позицию pos будут вставлены значения массива ins.

Пример: arr=[1,2,3,4,5] ins=[7,8,9] pos=3 результат: [1,2,3,7,8,9,4,5]

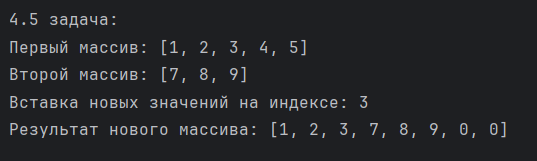
## Алгоритм решения

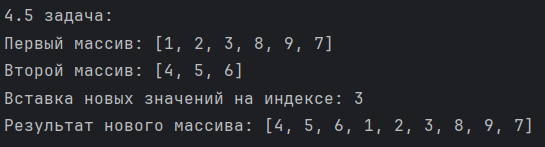
1. Создаем новый массив с увеличенной длиной на количество элементов массива ins.

2. Копируем все элементы из arr в новый массив.

3. Вставляем элементы из ins в позицию pos нового массива.

## Тестирование





# Задание 4.7

## Текст задания

Возвратный реверс.

Дана сигнатура метода: public int[] reverseBack (int[] arr); Необходимо реализовать метод таким образом, чтобы он возвращал новый массив, в котором значения массива arr записаны задом наперед.

Пример: arr=[1,2,3,4,5] результат: [5,4,3,2,1]

## Алгоритм решения

1. Создаем новый массив new\_arr с тем же размером, что и исходный arr.

2. Итерируем по элементам исходного массива от начала до конца.

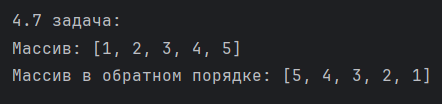
3. Для каждого элемента исходного массива:

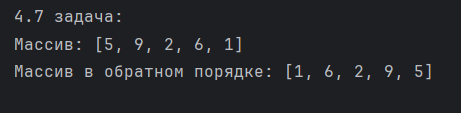
Присваиваем значение этого элемента соответствующему индексу в новом массиве.

Индекс для нового массива вычисляем как arr.length - 1 - i, где i - текущий индекс в исходном массиве.

4. Возвращаем новый массив new\_arr.

## Тестирование





# Задание 4.9

## Текст задания

Все вхождения.

Дана сигнатура метода: public int[] findAll (int[] arr, int x); Необходимо реализовать метод таким образом, чтобы он возвращал новый массив, в котором записаны индексы всех вхождений числа x в массив arr.

Пример: arr=[1,2,3,8,2,2,9] x=2 результат: [1,4,5]

## Алгоритм решения

Алгоритм решения задачи.

1. Создаем новый массив result для хранения индексов найденных значений.

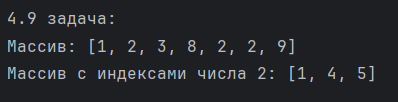
2. Итерируем по всем элементам входного массива arr.

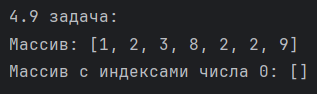
3. При каждой итерации проверяем, равен ли текущий элемент x.

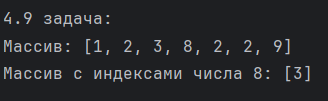
4. Если элемент равен x, добавить его индекс в массив result.

5. В конце возвращаем массив result.

## Тестирование







# Код программы

<https://github.com/AdelinaPs/Java_lab1/blob/master/src/Main.java>