|  |  |  |
| --- | --- | --- |
| 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 6 | | |
|  | | Работу выполнил  студент группы ПМИ-2/2023 2 курса  Волков А.Н.  «\_\_\_» \_\_\_\_\_\_\_\_2024 г. |
| Работу проверил  Ракина В.Д.  «\_\_\_» \_\_\_\_\_\_\_\_\_ 2024 г. |
| Пермь 2024 | | |

Оглавление

[Задание 1. Методы 5](#_Toc180473527)

[Текст задания 5](#_Toc180473528)

[2. Сумма знаков 5](#_Toc180473529)

[4. Есть ли позитив 5](#_Toc180473530)

[6. Большая буква 5](#_Toc180473531)

[8. Делитель 5](#_Toc180473532)

[10. Многократный вызов 5](#_Toc180473533)

[Алгоритм решения 6](#_Toc180473534)

[2. Сумма знаков 6](#_Toc180473535)

[4. Есть ли позитив 6](#_Toc180473536)

[6. Большая буква 6](#_Toc180473537)

[8. Делитель 6](#_Toc180473538)

[10. Многократный вызов 6](#_Toc180473539)

[Тестирование 7](#_Toc180473540)

[2. Сумма знаков 7](#_Toc180473541)

[4. Есть ли позитив 7](#_Toc180473542)

[6. Большая буква 7](#_Toc180473543)

[8. Делитель 7](#_Toc180473544)

[10. Многократный вызов 7](#_Toc180473545)

[Задание 2. Условия 8](#_Toc180473546)

[Текст задания 8](#_Toc180473547)

[2. Безопасное деление 8](#_Toc180473548)

[4. Строка сравнения 8](#_Toc180473549)

[6. Тройная сумма 8](#_Toc180473550)

[8. Возраст 8](#_Toc180473551)

[10. Вывод дней недели 8](#_Toc180473552)

[Алгоритм решения 9](#_Toc180473553)

[2. Безопасное деление 9](#_Toc180473554)

[4. Строка сравнения 9](#_Toc180473555)

[6. Тройная сумма 9](#_Toc180473556)

[8. Возраст 10](#_Toc180473557)

[10. Вывод дней недели 10](#_Toc180473558)

[Тестирование 10](#_Toc180473559)

[2. Безопасное деление 10](#_Toc180473560)

[4. Строка сравнения 10](#_Toc180473561)

[6. Тройная сумма 10](#_Toc180473562)

[8. Возраст 11](#_Toc180473563)

[10. Вывод дней недели 11](#_Toc180473564)

[Задание 3. Циклы 12](#_Toc180473565)

[Текст задания 12](#_Toc180473566)

[2. Числа наоборот 12](#_Toc180473567)

[4. Степень числа 12](#_Toc180473568)

[6. Одинаковость 12](#_Toc180473569)

[8. Левый треугольник 12](#_Toc180473570)

[10.Угадайка 12](#_Toc180473571)

[Алгоритм 13](#_Toc180473572)

[2. Числа наоборот 13](#_Toc180473573)

[4. Степень числа 13](#_Toc180473574)

[6. Одинаковость 13](#_Toc180473575)

[8. Левый треугольник 14](#_Toc180473576)

[10.Угадайка 14](#_Toc180473577)

[Тестирование 14](#_Toc180473578)

[2. Числа наоборот 14](#_Toc180473579)

[4. Степень числа 14](#_Toc180473580)

[6. Одинаковость 14](#_Toc180473581)

[8. Левый треугольник 15](#_Toc180473582)

[10.Угадайка 15](#_Toc180473583)

[Задание 4. Массивы 16](#_Toc180473584)

[Текст задания 16](#_Toc180473585)

[2. Поиск последнего значения 16](#_Toc180473586)

[4. Добавление в массив 16](#_Toc180473587)

[6. Реверс 16](#_Toc180473588)

[8. Объединение 16](#_Toc180473589)

[10. Удалить негатив 16](#_Toc180473590)

[Алгоритм 17](#_Toc180473591)

[2. Поиск последнего значения 17](#_Toc180473592)

[4. Добавление в массив 17](#_Toc180473593)

[6. Реверс 17](#_Toc180473594)

[8. Объединение 18](#_Toc180473595)

[10. Удалить негатив 18](#_Toc180473596)

[Тестирование 18](#_Toc180473597)

[2. Поиск последнего значения 18](#_Toc180473598)

[4. Добавление в массив 18](#_Toc180473599)

[6. Реверс 19](#_Toc180473600)

[8. Объединение 19](#_Toc180473601)

[10. Удалить негатив 19](#_Toc180473602)

[Ссылка на GitHub 20](#_Toc180473603)

# Задание 1. Методы

## Текст задания

### 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

### 4. 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

### 6. 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’

### 8. 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

### 10. 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.

## Алгоритм решения

### 2. Сумма знаков

1. Сложить числа из разряда единиц
2. Вернуть результат

### 4. Есть ли позитив

1. Проверить, является ли полученное число большим, чем 0
2. Вернуть true, если больше, иначе false

### 6. Большая буква

1. Проверить, является ли полученный символ заглавной буквой английского алфавита
2. Вернуть true, если является, иначе false

### 8. Делитель

1. Проверить, делится ли первое число на второе нацело или наоборот
2. Вернуть true, если делится, иначе false

### 10. Многократный вызов

1. Найти сумму чисел из разряда единиц первого и второго чисел
2. Найти сумму чисел из разряда единиц суммы первого и второго чисел и третьего числа
3. Считать, пока не сложим таким образом все 5 чисел
4. Вернуть результат

## Тестирование

### 2. Сумма знаков

### 4. Есть ли позитив

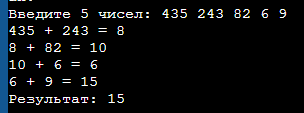
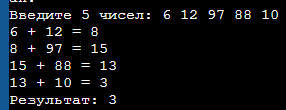
### 6. Большая буква

### 8. Делитель

### 10. Многократный вызов

# Задание 2. Условия

## Текст задания

### 2. 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

### 4. 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)

### 6. 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

### 8. 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

### 10. 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

## Алгоритм решения

### 2. Безопасное деление

1. Проверить, чему равен y. Если 0, то вернуть 0
2. Вернуть результат деления x на y

### 4. Строка сравнения

1. Сравнить полученные числа
2. Создать строку, состоящую из первого числа, знака равенства/неравенства и второго числа
3. Если первое число больше, то знак «>»
4. Если они равны, то «=»
5. Если второе число больше, то знак «<»
6. Вернуть полученную строку

### 6. Тройная сумма

1. Проверить, равна ли сумма первого и второго чисел третьему числу. Если равна, то вернуть true
2. Проверить, равна ли сумма второго и третьего чисел первому числу. Если равна, то вернуть true
3. Проверить, равна ли сумма третьего и первого чисел второму числу. Если да, то вернуть true
4. Иначе false

### 8. Возраст

1. Проверить, принадлежит ли полученное число диапазону [10; 14]. Если да, то вернуть «лет»
2. Проверить, принадлежит ли число из разряда единиц диапазону [5; 9] или {0}. Если да, то вернуть «лет»
3. Проверить, принадлежит ли число из разряда единиц {1}. Если да, то вернуть год
4. Проверить, принадлежит ли число из разряда единиц диапазону [2; 4]. Если да, то вернуть «года»

### 10. Вывод дней недели

1. Если полученная строка не является названием дня недели, то вернуть «Не является днём недели»
2. Вывести все дни недели, начиная с полученного до воскресенья

## Тестирование

### 2. Безопасное деление

### 4. Строка сравнения

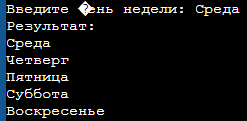
### 6. Тройная сумма

### 8. Возраст

### 10. Вывод дней недели

# Задание 3. Циклы

## Текст задания

### 2. 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)

### 4. 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

### 6. Одинаковость

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

### 8. Левый треугольник

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

### 10.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

## Алгоритм

### 2. Числа наоборот

1. Создать цикл, который будет идти количество итераций, равное полученному числу с прибавленной единицей
2. На каждой итерации сохранять в строку числа на единицу меньше, чем предыдущее
3. Вернуть строку

### 4. Степень числа

1. Создать цикл, количество итераций которого будет равно второму полученному числу
2. В первой итерации единица умножается на первое число, в следующих итерациях число, полученное в предыдущей итерации, умножается на первое число
3. Вернуть число, полученное в последней итерации

### 6. Одинаковость

1. «Поднять» флажок, который будет отвечать за одинаковость числа
2. Начиная со второй, сравнивать каждую цифру с предыдущей.
3. Сравнивать, пока не пройдём всё число
4. Если будет хоть одно неравенство, то опустить флажок
5. Если флажок поднят, то вернуть true, иначе false

### 8. Левый треугольник

1. Создать цикл, количество итераций которого будет равно полученному числу
2. Выводить количество звёздочек, равное номеру итерации

### 10.Угадайка

1. Сгенерировать случайное число в диапазоне [0; 9]
2. Вводить числа, пока не введённое число не будет равно сгенерированному

## Тестирование

### 2. Числа наоборот





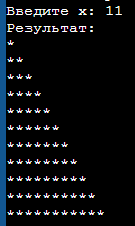
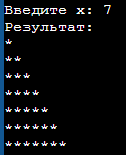
### 4. Степень числа



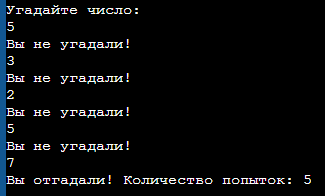
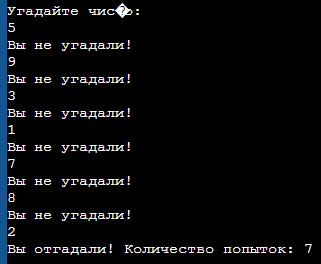
### 6. Одинаковость



### 8. Левый треугольник



### 10.Угадайка



# Задание 4. Массивы

## Текст задания

### 2. 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

### 4. 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

### 6. 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

### 8. 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)

### 10. 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

## Алгоритм

### 2. Поиск последнего значения

1. Создать временную переменную, которая будет хранить позицию последнему элемента, равному полученному числу. По умолчанию равняется -1
2. Пройтись по массиву слева направо
3. Если элемент массива равен полученном числу, то во временную переменную записывается позиция этого элемента
4. Вернуть временную переменную

### 4. Добавление в массив

1. Создать массив, длинна которого будет на единицу больше, чем полученный массив
2. Записать в новый массив все значения из старого до номера полученной позиции
3. Записать в новый массив полученное число на номер полученной позиции
4. Записать в новый массив оставшиеся числа из полученного массива после номера полученной позиции
5. Вернуть массив

### 6. Реверс

1. Дойти до середины массива, меняя числа первое с последним, второе с предпоследним и т.д.
2. Вернуть массив

### 8. Объединение

1. Создать массив, длинна которого будет равна сумме длин первого и второго массивов
2. Записать в новый массив числа из первого массива
3. Записать в новый массив числа из второго массива, начиная с позиции, равной длине первого массива
4. Вернуть новый массив

### 10. Удалить негатив

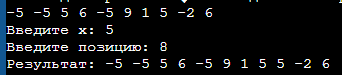
1. Посчитать количество отрицательных чисел в полученном массиве
2. Создать новый массив, длина которого будет равна длине полученного массива минус количество отрицательных чисел
3. Записать в новый массив только неотрицательные числа
4. Вернуть новый массив

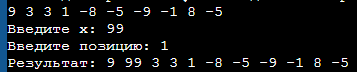
## Тестирование

### 2. Поиск последнего значения



### 4. Добавление в массив





### 6. Реверс





### 8. Объединение



### 10. Удалить негатив







# Ссылка на GitHub

<https://github.com/kuv4lda/PSU_Java/blob/main/lab1/Main.java>