|  |  |  |
| --- | --- | --- |
| 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 ИТХ-1,2-2023 2 курса  Гилев А.А.  «8» Октября\_2024 г. |
| Работу проверил  Фамилия И.О. преподавателя  Ракина В.Д.  «9» Октября\_ 2024 г. |
| Пермь 2024 | | |

СОДЕРЖАНИЕ

[Задание 1 1](#_Toc179326484)

[Текст задания 1](#_Toc179326485)

[Алгоритм решения 1](#_Toc179326486)

[Тестирование 1](#_Toc179326487)

[Код программы 1](#_Toc179326488)

[Задание 2 1](#_Toc179326489)

[Текст задания 1](#_Toc179326490)

[Алгоритм решения 1](#_Toc179326491)

[Тестирование 1](#_Toc179326492)

[Код программы 1](#_Toc179326493)

[Задание 3 1](#_Toc179326494)

[Текст задания 1](#_Toc179326495)

[Алгоритм решения 1](#_Toc179326496)

[Тестирование 1](#_Toc179326497)

[Код программы 1](#_Toc179326498)

[Задание 4 1](#_Toc179326499)

[Текст задания 1](#_Toc179326500)

# Задание 1

## Текст задания

Задание 1 состоит из реализации 5 методов на языке C#:

1. 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

2. 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

3. 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

4. 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

5. 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

Для начала я реализовал первичный выбор задания, использовав оператор выбора switch в пределах бесконечного цикла, чтобы у пользователя была возможность воспользоваться разными функциями, без перезапуска программы. Для выхода было выбрано значение 0, а при вводе неверного значения, программа оповещает пользователя об этом и просит его ввести значение заново. После выбора задания, программа запускает метод void pupupu(x) для дальнейшего выбора конкретного метода и ввода начальных данных.

Первый метод, как и все, кроме пятого, в первом задании, состоит из одного оператора return. В качестве аргумента служит выражение x%10 \* (x/10)% перемножающее последние две цифры изначального числа и сразу же возвращающее их в метод pupupu, через который они выводятся на экран.

Второй метод абсолютно аналогичен первому, но в качестве аргумента содержит логическое выражение x >= 0 и поэтому значение так же возвращается логическое.

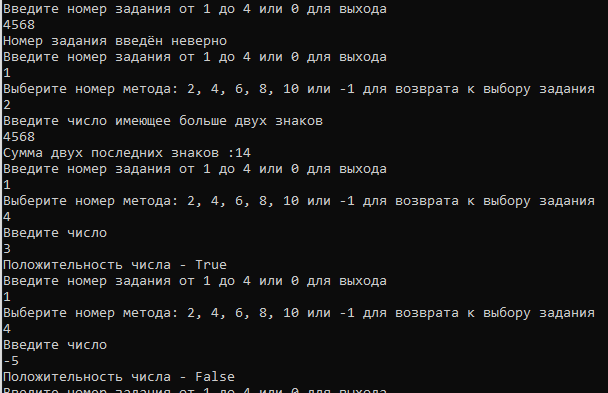
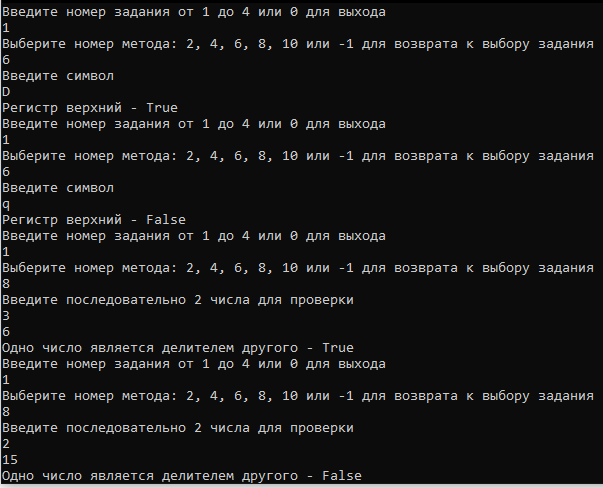
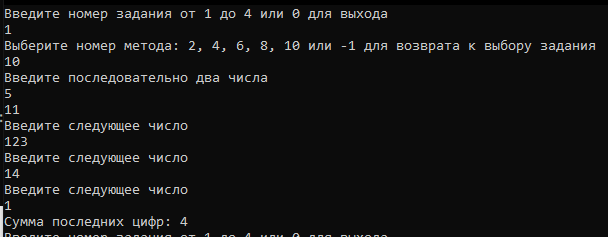
Третий метод использует встроенный метод структуры char – IsUpper(x), проверяющий символ на регистр и возвращающий логическое значение аналогично предыдущим методам.

Четвёртый метод в качестве аргумента оператора return имеет сложное логическое выражение : a % b == 0 || b % a == 0 которое проверяет сразу два случая: делиться ли первое число на второе без остатка, делится ли второе число на первое без остатка, а так же и оба этих условия одновременно, возвращая значение «true» в такой ситуации.

Пятый метод не использует дополнительных переменных. Сначала он вычисляет сумму остатков от деления двух данных чисел, записывая результат в ячейку памяти одной из них, после чего просит пользователя ввести следующее число. От текущего результата вычисляется остаток от деления на 10 и складывается с остатком от деления на 10 от нового данного числа и эта процедура повторяется ещё 3 раза, в конечном итоге вычисляя сумму разрядов единиц 6 чисел и возвращая её в метод pupupu для дальнейшего вывода на экран.

Номера строк в коде: 36-78, 214-245

## Тестирование

## Код программы

***Далее***

# Задание 2

## Текст задания

1. 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

2. 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”

3. 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

4. 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”

5. 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

Алгоритм решения задачи.

## Тестирование

Скриншоты результата работы программы.

## Код программы

Ссылка на githab

# Задание 3

## Текст задания

1. 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”

2. 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

3. 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

4. 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:

\*

\*\*

\*\*\*

\*\*\*\*

5. 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

Алгоритм решения задачи.

## Тестирование

Скриншоты результата работы программы.

## Код программы

Ссылка на githab

# Задание 4

## Текст задания

1. 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

2. 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]

3. 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]

4. 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]

5. 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

Алгоритм решения задачи.

## Тестирование

Скриншоты результата работы программы.

## Код программы

Ссылка на github: <https://github.com/NapstaTT/C-LAB1/blob/main/Program.cs>