## **Министерство Образования и Исследований**

## **Республики Молдова**

## **Технический Университет Молдовы**

## Факультет Вычислительной Техники, Информатики и Микроэлектроники

## **Отчет**

## по лабораторной работе Nr.1. по:

## **Декларативному Программированию**

## 

## 

## Выполнила ст.гр. TI-219:Галупа Яна

## Проверил: Лях Аркадий

## 

## Кишинев – 2023

**Тема: Использование Python**

# **Задания к самостоятельной работе**

### **Задание 1**

Напишите код по следующему словесному алгоритму:

1. Попросить пользователя ввести число от 1 до 9. Полученные данные связать с переменной x.
2. Если пользователь ввел число от 1 до 3 включительно, то …

* попросить пользователя ввести строку. Полученные данные связать с переменной s;
* попросить пользователя ввести число повторов строки. Полученные данные связать с переменной n, предварительно преобразовав их в целочисленный тип;
* выполнить цикл повторения строки n раз;
* вывести результат работы цикла.

1. Если пользователь ввел число от 4 до 6 включительно, то …

* попросить пользователя ввести степень, в которую следует возвести число. Полученные данные связать с переменной m;
* реализовать возведение числа x в степень m;
* вывести полученный результат.

1. Если пользователь ввел число от 7 до 9, то выполнить увеличения числа x на единицу в цикле 10 раз, при этом на экран вывести все 10 чисел.
2. Во всех остальных случаях выводить надпись "Ошибка ввода".

### **Задание 2**

Напишите программу, которая бы выполняла следующие задачи:

1. выводила название программы "Общество в начале XXI века";
2. запрашивала у пользователя его возраст;
3. если пользователь вводит числа от 0 до 7, то программа выводила надпись "Вам в детский сад";
4. от 7 до 18 - "Вам в школу";
5. от 18 до 25 - "Вам в профессиональное учебное заведение";
6. от 25 до 60 - "Вам на работу";
7. от 60 до 120 – "Вам предоставляется выбор";
8. меньше 0 и больше 120 – пятикратный вывод надписи "Ошибка! Это программа для людей!"

В программе желательно использовать все "атрибуты" структурного программирования: ветвление и цикл.

# Задание 1

# 1

x = int(input('Введите число от 1 до 9:\n'))

# 2

if 1 <= x <= 3:

s = input("Введите строку:\n")

n = int(input("Введите число повторов:\n"))

for \_ in range(n):

print(s)

# 3

elif 4 <= x <= 6:

m = int(input("Введите степень, в которую возвести число\n"))

print(pow(x, m))

# 4

elif 7 <= x <= 9:

for \_ in range(10):

x += 1

print(x)

else:

print('Ошибка ввода')

# Задание 2

# 1

print('Общество в начале XXI века')

# 2

a = int(input('Введите свой возраст:\n'))

# 3

if 0 <= a <= 7:

print('Вам в детский сад')

# 4

elif 7 <= a <= 18:

print('Вам в школу')

# 5

elif 18 <= a <= 25:

print('Вам в профессианальное учебное заведение')

# 6

elif 25 <= a <= 60:

print('Вам на работу')

# 7

elif 60 <= a <= 120:

print('Вам предоставляется выбор')

# 8

elif 0 >= a >= 120:

for \_ in range(5):

print('Ошибка! Это программа для людей!')

В ходе выполнения лабораторной работы были успешно применены основные принципы структурного программирования для решения двух задач. В работе использовались ветвления (`if`, `elif`, `else`) для создания алгоритмов, способных принимать разные ветви исполнения в зависимости от условий. Это позволило эффективно управлять ходом программы в зависимости от входных данных.

Также в лабораторной работе были успешно применены циклы (`for`, `while`), что дало возможность создавать повторяющиеся блоки кода для обработки данных или выполнения определенных операций несколько раз. Это привнесло в программу гибкость и позволило избежать дублирования кода.

В целом, выполнение лабораторной работы подчеркнуло важность понимания и применения структурного программирования для создания эффективных и легко читаемых программ. Это также предоставило отличную возможность применить полученные знания на практике.