РЕСПУБЛИКА БЕЛАРУСЬ

КОМИТЕТ ПО ОБРАЗОВАНИЮ МИНГОРИСПОЛКОМА учреждение образования

МИНСКИЙ ГОСУДАРСТВЕННЫЙ КОЛЛЕДЖ ЦИФРОВЫХ

ТЕХНОЛОГИИ

Группа 87 ТП

Отчет

по лабораторной работе №4 “Разработка и отладка алгоритмов, содержащих циклы”

Учебный предмет

«Инструментальное программное обеспечение»

Исполнитель: Ковальчук Р.Ю

Руководитель: Бровка Д.С.

Минск, 2025

Цель:

1. Ознакомление с основами работы циклов в Python.
2. Изучение генераторов списков для преобразования данных.
3. Развитие практических навыков решения задач с использованием циклов и генераторов.

**Задачи лабораторной работы:**

1. Изучить синтаксис циклов for и while.
2. Реализовать примеры перебора и обработки списков.
3. Научиться использовать генераторы списков для создания и фильтрации данных.
4. Выполнить практические задачи, включая создание нового списка из заданного диапазона чисел.

# Ход работы

1. Изучите теоретические сведения
2. Выполните общие и индивидуальные задания в тетради для лабораторных и практических работ или на рабочем компьютере
3. Сдайте выполненные задание в Classroom или сдать тетрадь преподавателю
4. Защитить выполненные задания

Контрольные вопросы:

1. Что такое цикл for в Python и когда его следует использовать?  
Цикл for повторяет действия для каждого элемента последовательности.  
Используется, когда известно количество повторений.  
Пример:  
for i in range(5):  
print(i)  
2. Объясните, как работает цикл while. В каких ситуациях он более  
предпочтителен, чем цикл for?  
Цикл while выполняется, пока условие истинно. Используется, когда заранее  
неизвестно, сколько раз нужно повторить действия.  
Пример:  
i = 0  
while i &lt; 3:  
print(i)  
  
i += 1  
3. Какой синтаксис используется для создания генераторов списков в  
Python? Приведите пример.  
Синтаксис: [выражение for элемент in последовательность if условие]  
Пример:  
nums = [x\*\*2 for x in range(1, 6)]  
4. Как можно использовать оператор if вместе с циклами, чтобы  
фильтровать данные в списках?  
Оператор if позволяет выбирать только те элементы, которые удовлетворяют  
условию.  
Пример:  
x = [n for n in range(10) if n % 2 == 0]  
print (x)  
5. Чем отличается генератор списка от обычного цикла for, который  
заполняет список поэтапно?  
Генератор списка — короткая, быстрая запись в одну строку.  
Обычный for требует больше строк, но удобен для сложной логики.  
6. Как можно использовать функции input() и print() в сочетании с  
циклами в Python?  
input() — для ввода.  
print() — для вывода результатов в каждом повторении.  
Пример:  
for i in range(3):  
name = input(&quot;Имя: &quot;)  
print(&quot;Привет,&quot;, name)  
7. Что произойдет, если условие в цикле while никогда не ложно? Как это  
называется и как его избежать?  
Цикл не остановится — получится бесконечный цикл.  
Избежать можно, изменяя переменные внутри цикла, чтобы условие стало  
ложным.  
  
8. Объясните, как использовать функцию len() в сочетании с циклами  
для работы со списками.  
len() показывает длину списка.  
Её используют, чтобы проходить список по индексам.  
Пример:  
nums = [10, 20, 30]  
for i in range(len(nums)):  
print(nums[i])  
9. Как можно объединить генераторы списков с условными  
выражениями, чтобы создать новый список, удовлетворяющий  
определенным условиям?  
Добавляют if в генератор, чтобы отбирать элементы по условию.  
Пример:  
result = [x for x in range(10) if x &gt; 5]  
10. Какие возможные ошибки могут возникнуть при написании циклов  
и генераторов списков, и как их можно избежать?  
Ошибки: бесконечный цикл, неправильное условие, опечатки в переменных,  
неверные типы данных.  
Избежать можно с помощью проверки условий и от