РЕСПУБЛИКА БЕЛАРУСЬ

КОМИТЕТ ПО ОБРАЗОВАНИЮ МИНГОРИСПОЛКОМА

учреждение образования

МИНСКИЙ ГОСУДАРСТВЕННЫЙ КОЛЛЕДЖ ЦИФРОВЫХ ТЕХНОЛОГИИ

##### Группа 82

## Отчет

## по лабораторной работе № 3

# "**Разработка и отладка разветвляющихся алгоритмов**

"

Учебный предмет

«Инструментальное программное обеспечение»

Исполнитель: Белоусов И.Р.

Руководитель: Бровка Д.С.

Минск, 2024

# Цели и задачи

1. Изучить принципы разработки разветвляющихся алгоритмов.
2. Освоить применение операторов ветвления в языке программирования Python.
3. Научиться отлаживать программы с условными конструкциями.

# 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. Оператор `if` в Python: Оператор `if` используется для условного выполнения кода. Он проверяет, истинно ли заданное условие. Если условие истинно, то код внутри блока `if` выполняется. В противном случае, он пропускается.  
  
 

2. Роль двоеточия `:` в условных конструкциях: Двоеточие после условия в `if` (и `elif`, `else`) указывает на начало блока кода, который будет выполняться, если условие истинно. В Python отступы определяют границы блоков кода, а двоеточие сигнализирует о начале такого блока.  
  
3. Логические операторы в Python:  
  
   \* `and`: Логическое "И". Выражение истинно, только если оба операнда истинны.  
   \* `or`:   Логическое "ИЛИ". Выражение истинно, если хотя бы один операнд истинен.  
   \* `not`: Логическое "НЕ". Инвертирует значение операнда (истинно становится ложно, и наоборот).  
  
4. Блок кода в Python: Блок кода – это группа инструкций, объединенных логически и выполняющихся вместе. В Python границы блока кода определяются отступами (обычно 4 пробела). Все строки с одинаковым отступом принадлежат одному блоку.  
  
5. Логические значения в Python: В Python логические значения представляются булевыми типами данных:  
  
   \* `True`: Истина.  
   \* `False`: Ложь.  
  
6. Оператор `elif`: `elif` (сокращение от "else if") используется для проверки дополнительных условий после `if`. Если условие в `if` ложно, Python проверяет условия в `elif` по порядку, до тех пор, пока не найдет истинное условие или не дойдет до `else`.

7. Функция оператора `else`: `else` выполняет блок кода, если все предыдущие условия (`if` и `elif`) оказались ложными. `else` является необязательной частью условной конструкции.  
  
8. Операторы сравнения в Python:  
  
   \* `==`: Равно.  
   \* `!=`: Не равно.  
   \* `>`: Больше.  
   \* `<`: Меньше.  
   \* `>=`: Больше или равно.  
   \* `<=`: Меньше или равно.  
  
9. Проверка равенства и неравенства: В Python равенство проверяется с помощью оператора `==`, а неравенство – с помощью оператора `!=`.  
  
10. Операторы сравнения "больше", "меньше", "больше или равно", "меньше или равно": Они обозначаются символами `>`, `<`, `>=`, `<=` соответственно.