**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ,**

**СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА» (СПбГУТ)**

Факультет Инфокоммуникационных сетей и систем Кафедра Защищенных систем связи

Дисциплина Защищенные операционные системы

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ

Циклы

*(тема отчета)*

Направление/специальность подготовки

10.03.01 Информационная безопасность

*(код и наименование направления/специальности)*

Студент:

Емельянов Н.Р. ИКБ-32

*(Ф.И.О., № группы) (подпись)*

Михайлов И.Д. ИКБ-32

*(Ф.И.О., № группы) (подпись)*

Преподаватель:

Смирнов Д.Н.

*(Ф.И.О. преподавателя) (подпись)*

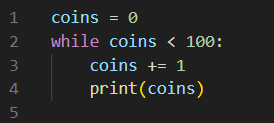
**Цель**

На практическом примере разобраться в работе циклов в python и оптимизировать код при помощи циклов, во избежание дублирования кода.

**Задача**

Написать код с использованием циклов, который будет прибавлять по 1 монете, пока их не станет ровно 100.

**Ход выполнения**



Создаем переменную «**coins**» и задаем ей начальное значение 0, что равно начальному количеству монет.

Используем цикл **while**, который будет проверять текущее значение переменной «coins» и если оно меньше **100**, то прибавляет к этому значению 1, пока оно не будет равно 100.

**Вывод**

Цикл **while** в python позволяет нам оптимизировать код и избежать дублирования систематически повторяющихся действий. На данном примере, вместо того, чтобы расписывать код, прибавляя по 1 монете в каждой строчке (**coins = coins + 1**), мы использовали цикл **while** и успешно выполнили задачу за меньшее время.