МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФГАОУ ВО «СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

ИНСТИТУТ ИНФОРМАЦИОННЫХ

ТЕХНОЛОГИЙ И ТЕЛЕКОММУНИКАЦИЙ

КАФЕДРА ПРИКЛАДНОЙ ИНФОРМАТИКИ

**Лабораторная работа № 1**

по дисциплине

«Информационные технологии и программирования»

**Выполнил:**

Попов Антон Андреевич

Студент 1 курса группы ПИН-б-о-22-1

Направления подготовки

09.03.03 Прикладная информатика

очной формы обучения

Ставрополь, 2023 г.

# Тема: Основы объектно-ориентированного программирования на ЯП Python

**Цель работы:** изучить базовые понятия (классы, подклассы и методы). Реализовать фундаментальные принципы объектно-ориентированного программирования.

**Ход работы**

Вариант - 17

№ 4.3.1 Римское число. Листинг приведён в файлах:  
 [main.py](https://github.com/Antrt/-/blob/main/pytonLR1/oop_lr01_task_04_03_01/main.py) [roman.py](https://github.com/Antrt/-/blob/main/pytonLR1/oop_lr01_task_04_03_01/roman.py) [test\_roman.py](https://github.com/Antrt/-/blob/main/pytonLR1/oop_lr01_task_04_03_01/%D0%A2%D0%B5%D1%81%D1%82%D1%8B/test_roman.py)

Также приведена [UML-диаграмма](https://github.com/Antrt/-/blob/main/pytonLR1/oop_lr01_task_04_03_01/uml.png) проекта.

В модуле *roman.py* были доработаны несколько методов у класса *Roman*, включая обработку некорректных входных данных. В основном приложении реализован простой тест работоспособности класса *Roman.*

№ 4.3.2 Пицерия. Листинг приведён в файлах:  
 [main.py](https://github.com/Antrt/-/blob/main/pytonLR1/oop_lr01_task_04_03_02/main.py) [заказ.py](https://github.com/Antrt/-/blob/main/pytonLR1/oop_lr01_task_04_03_02/%D0%B7%D0%B0%D0%BA%D0%B0%D0%B7.py) [пицца.py](https://github.com/Antrt/-/blob/main/pytonLR1/oop_lr01_task_04_03_02/%D0%BF%D0%B8%D1%86%D1%86%D0%B0.py) [терминал.py](https://github.com/Antrt/-/blob/main/pytonLR1/oop_lr01_task_04_03_02/%D1%82%D0%B5%D1%80%D0%BC%D0%B8%D0%BD%D0%B0%D0%BB.py) test\_заказ.py  
 test\_пицца.py  
 [test\_терминал.py](https://github.com/EGP24/oop/blob/main/pythonLR1/oop_lr01_task_04_03_02/tests/test_%D1%82%D0%B5%D1%80%D0%BC%D0%B8%D0%BD%D0%B0%D0%BB.py)

Также приведена [UML-диаграмма](https://github.com/EGP24/oop/blob/main/pythonLR1/oop_lr01_task_04_03_02/uml.png) проекта.

Реализованы модули *заказ.ру, пицца.ру, терминал.ру.* В их классах были реализованы методы и связи между друг другом. В основном приложении реализован алгоритм пиццерии, указанный в задании.

№ 4.3.3 Банковские вклады. Листинг приведён в файлах:  
 [main.py](https://github.com/Antrt/-/blob/main/pytonLR1/oop_lr01_task_04_03_03/main.py) [deposit.py](https://github.com/Antrt/-/blob/main/pytonLR1/oop_lr01_task_04_03_03/deposit.py) [test\_time\_deposit.py](https://github.com/Antrt/-/blob/main/pytonLR1/oop_lr01_task_04_03_03/%D0%A2%D0%B5%D1%81%D1%82%D1%8B/test_time_deposit.py) [test\_bonus\_time\_deposit.py](https://github.com/Antrt/-/blob/main/pytonLR1/oop_lr01_task_04_03_03/%D0%A2%D0%B5%D1%81%D1%82%D1%8B/test_bonus_time_deposit.py) [test\_compound\_time\_deposit.py](https://github.com/Antrt/-/blob/main/pytonLR1/oop_lr01_task_04_03_03/%D0%A2%D0%B5%D1%81%D1%82%D1%8B/test_compound_time_deposit.py)

Также приведена [UML-диаграмма](https://github.com/Antrt/-/blob/main/pytonLR1/oop_lr01_task_04_03_03/uml.png) проекта.

Реализован базовый класс *TimeDeposit,* на его основе были реализованы ещё два класса: *BonusTimeDeposit* и *CompoundTimeDeposit.* В основном приложении реализован алгоритм для вычисления прибыли от каждого из вкладов на заданный срок и заданную сумму.

№ 4.3.4 Простой класс. Листинг приведён в файлах:  
 [main.py](https://github.com/Antrt/-/blob/main/pytonLR1/oop_lr01_task_04_03_04/main.py) [stack.py](https://github.com/Antrt/-/blob/main/pytonLR1/oop_lr01_task_04_03_04/main.py) [test\_stack.py](https://github.com/Antrt/-/blob/main/pytonLR1/oop_lr01_task_04_03_04/%D0%A2%D0%B5%D1%81%D1%82%D1%8B/test_stack.py)

Также приведена [UML-диаграмма](https://github.com/Antrt/-/blob/main/pytonLR1/oop_lr01_task_04_03_04/uml.png) проекта.

Реализован класс *Stack* с основными его базовыми методами.

№ 4.3.5 Класс-контейнер. Листинг приведён в файлах:  
 [main.py](https://github.com/Antrt/-/blob/main/pytonLR1/oop_lr01_task_04_03_05/main.py) [stack\_collection.py](https://github.com/Antrt/-/blob/main/pytonLR1/oop_lr01_task_04_03_05/stack_collection.py) [test\_stack\_collection.py](https://github.com/Antrt/-/blob/main/pytonLR1/oop_lr01_task_04_03_05/%D0%A2%D0%B5%D1%81%D1%82%D1%8B/test_stack_collection.py)Также приведена [UML-диаграмма](https://github.com/Antrt/-/blob/main/pytonLR1/oop_lr01_task_04_03_05/uml.png) проекта.

Реализован класс *StackCollection* на основе класса *Stack,* а также ряд основных методов для взаимодействия с коллекцией математических промежутков, таких как добавление/удаление/получение элемента, json импорт/экспорт, сравнение коллекций и другие методы.

№ 4.3.6 Иерархия классов. Листинг приведён в файлах:  
 [main.py](https://github.com/Antrt/-/blob/main/pytonLR1/oop_Lr01_task_04_03_06/main.py) [writing\_affiliation.py](https://github.com/Antrt/-/blob/main/pytonLR1/oop_Lr01_task_04_03_06/writing_affiliation.py) [pencil.py](https://github.com/Antrt/-/blob/main/pytonLR1/oop_Lr01_task_04_03_06/pencil.py) [pen.py](https://github.com/Antrt/-/blob/main/pytonLR1/oop_Lr01_task_04_03_06/pen.py) [gel\_pen.py](https://github.com/Antrt/-/blob/main/pytonLR1/oop_Lr01_task_04_03_06/gel_pen.py) [test\_writing\_affiliation.py](https://github.com/Antrt/-/blob/main/pytonLR1/oop_Lr01_task_04_03_06/%D0%A2%D0%B5%D1%81%D1%82%D1%8B/test_wrirting_affiliation.py) [test\_pencil.py](https://github.com/Antrt/-/blob/main/pytonLR1/oop_Lr01_task_04_03_06/%D0%A2%D0%B5%D1%81%D1%82%D1%8B/test_pencil.py) [test\_pen.py](https://github.com/Antrt/-/blob/main/pytonLR1/oop_Lr01_task_04_03_06/%D0%A2%D0%B5%D1%81%D1%82%D1%8B/test_pen.py) [test\_gel\_pen.py](https://github.com/Antrt/-/blob/main/pytonLR1/oop_Lr01_task_04_03_06/%D0%A2%D0%B5%D1%81%D1%82%D1%8B/test_gel_pen.py)Также приведена [UML-диаграмма](https://github.com/EGP24/oop/blob/main/pythonLR1/oop_lr01_task_04_03_06/uml.png) проекта.

Реализован базовый класс *WritingAffiliation,* а также на его основе классы *Pen, Pencil, GelPen.* Выстроена иерархия наследования и реализованы показательные примеры методов.

Ссылка на [репозиторий](https://github.com/Antrt/-/tree/main/pytonLR1), содержащий полностью выполненные задания.

**Вывод:** изучил базовые понятия (классы, подклассы и методы) Реализовал фундаментальные принципы объектно-ориентированного программирования.