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:**

Иванов Даниил Евгеньевич

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.

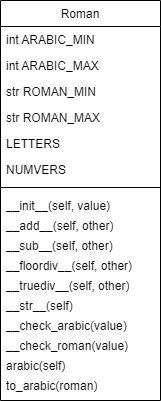
Выполнение работы:

4.3.1. Римское число

# Программирование на языке высокого уровня (Python).  
# Задание №4.3.1  
#  
# Выполнил: Иванов Даниил Евгеньевич

# Группа: ПИН-б-о-22-1

# E-mail: daniilivanov752@gmail.com  
  
  
from roman import Roman  
  
if \_\_name\_\_ == "\_\_main\_\_":  
  
 r1 = Roman("X")  
 r2 = Roman(5)  
  
 print(" Числа:", r1, r2, r1.arabic, r2.arabic)  
 print(" Сумма:", r1 + r2)  
 print(" Разность:", r1 - r2)  
 print("Произведение:", r1 \* r2)  
 print(" Частное:", r1 // r2)  
  
 print("\nПреобразование без создания объекта:")  
 print(2016, "=", Roman.to\_roman(2016))  
 print("MMXVI", "=", Roman.to\_arabic("MMXVI"))  
  
# -------------  
# Пример вывода:  
  
# Числа: X V 10 5  
# Сумма: XV  
# Разность: V  
# Произведение: L  
# Частное: II  
#  
# Преобразование без создания объекта:  
# 2016 = MMXVI  
# MMXVI = 2016

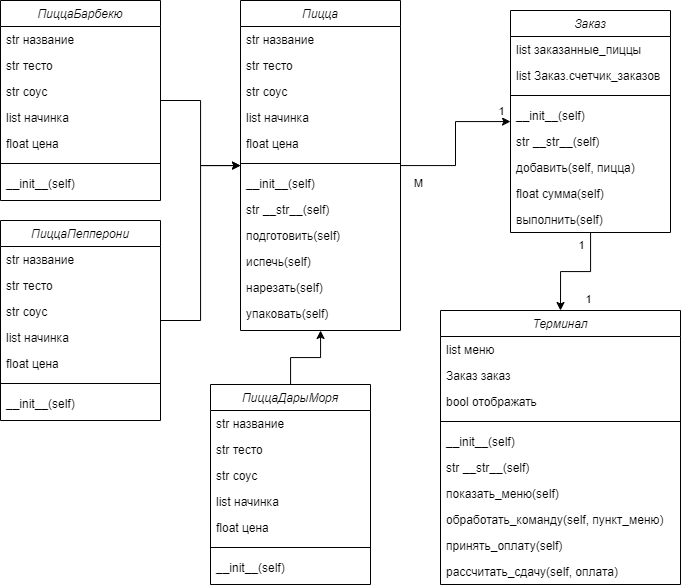


4.3.2. Пицерия

# Программирование на языке высокого уровня (Python).  
# Задание №4.3.2  
#  
# Выполнил: Иванов Даниил Евгеньевич

# Группа: ПИН-б-о-22-1

# E-mail: daniilivanov752@gmail.com  
  
  
from терминал import Терминал  
  
if \_\_name\_\_ == "\_\_main\_\_":  
  
 терминал1 = Терминал()  
 print(терминал1)  
 while True:  
 терминал1.показать\_меню()  
 пункт\_меню = input()  
 терминал1.обработать\_команду(пункт\_меню)  
  
# -------------  
# Пример вывода:  
  
# --------------------------------------------------  
# Пиццерия #1  
# Добро пожаловать!  
#  
# Меню:  
# 1. Пицца: Пепперони | Цена: 350.00 р.  
# Тесто: тонкое Соус: томатный  
# Начинка: пепперони, сыр моцарелла  
# 2. Пицца: Барбекю | Цена: 450.00 р.  
# Тесто: тонкое Соус: барбекю  
# Начинка: бекон, ветчина, зелень, сыр моцарелла  
# 3. Пицца: Дары моря | Цена: 550.00 р.  
# Тесто: пышное Соус: тар-тар  
# Начинка: кальмары, креветки, мидии, сыр моцарелла  
# Для выбора укажите цифру через <ENTER>.  
# Для отмены заказа введите -1  
# Для подтверждения заказа введите 0  
#  
# 1  
# Пицца Пепперони добавлена!  
# 2  
# Пицца Барбекю добавлена!  
# 0  
# Заказ подтвержен.  
# Заказ №2  
# 1. Пицца: Пепперони | Цена: 350.00 р.  
# Тесто: тонкое Соус: томатный  
# Начинка: пепперони, сыр моцарелла  
# 2. Пицца: Барбекю | Цена: 450.00 р.  
# Тесто: тонкое Соус: барбекю  
# Начинка: бекон, ветчина, зелень, сыр моцарелла  
# Сумма заказа: 800.00 р.  
# Введите сумму: 1000  
# Вы внесли 1000.00 р. Сдача: 200.00 р.  
#  
# Заказ поступил на выполнение...  
# 1. Пепперони  
# Начинаю готовить пиццу Пепперони  
# - замешиваю тонкое тесто...  
# - добавляю соус: томатный...  
# - и, конечно: пепперони, сыр моцарелла...  
# Выпекаю пиццу... Готово!  
# Нарезаю на аппетитные кусочки...  
# Упаковываю в фирменную упаковку и готово!  
# 2. Барбекю  
# Начинаю готовить пиццу Барбекю  
# - замешиваю тонкое тесто...  
# - добавляю соус: барбекю...  
# - и, конечно: бекон, ветчина, зелень, сыр моцарелла...  
# Выпекаю пиццу... Готово!  
# Нарезаю на аппетитные кусочки...  
# Упаковываю в фирменную упаковку и готово!  
#  
# Заказ №2 готов! Приятного аппетита!

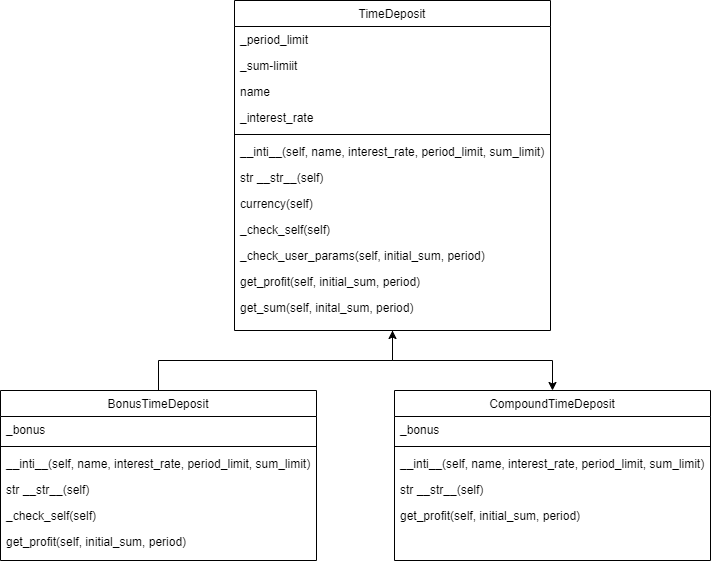


4.3.3. Банковские вклады

# Программирование на языке высокого уровня (Python).  
# Задание №4.3.3  
#  
# Выполнил: Иванов Даниил Евгеньевич

# Группа: ПИН-б-о-22-1

# E-mail: daniilivanov752@gmail.com  
  
from deposit import deposits  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 print("Добро пожаловать в систему подбора вкладов!")  
  
 while True:  
 print("\n-----")  
 print("Нажмите 1, чтобы подобрать вклад, или что угодно для выхода.")  
  
 answer = input()  
 if answer == "1":  
  
 initial\_sum = float(input("1/2: Введите начальную сумму вклада: "))  
 period = int(input("2/2: Введите срок вклада (мес.): "))  
  
 matched\_deposits = []  
 for deposit in deposits:  
 try:  
 deposit.\_check\_user\_params(initial\_sum, period)  
 matched\_deposits.append(deposit)  
 except AssertionError as err:  
 pass  
  
 if len(matched\_deposits) > 0:  
 print("{0:18} | {1:13} | {2:13}".format(  
 "Вклад", "Прибыль", "Итоговая сумма"  
 ))  
 for deposit in matched\_deposits:  
 print("{0:18} | {1:8,.2f} {3:4} | {2:8,.2f} {3:4}".format(  
 deposit.name,  
 deposit.get\_profit(initial\_sum, period),  
 deposit.get\_sum(initial\_sum, period),  
 deposit.currency))  
 else:  
 print("К сожалению, нет подходящих Вам вкладов.")  
  
 else:  
 break  
  
 print("\nСпасибо, что воспользовались терминалом банка! До встречи!")  
  
# -------------  
# Пример вывода (файл):  
#  
# Добро пожаловать в систему подбора вкладов!  
#  
# -----  
# Нажмите 1, чтобы подобрать вклад, или что угодно для выхода.  
# 1  
# 1/2: Введите начальную сумму вклада: 1000  
# 2/2: Введите срок вклада (мес.): 12  
# Вклад | Прибыль | Итоговая сумма  
# Сохраняй | 50.00 руб. | 1,050.00 руб.  
# Бонусный | 50.00 руб. | 1,050.00 руб.  
# С капитализацией | 51.16 руб. | 1,051.16 руб.



4.3.4. Простой класс

# Программирование на языке высокого уровня (Python).  
# Задание №4.3.4 Вариант 7  
#  
  
# Выполнил: Иванов Даниил Евгеньевич

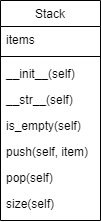
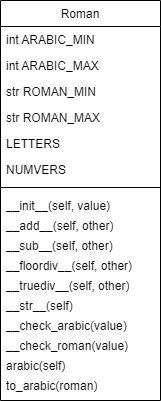
# Группа: ПИН-б-о-22-1

# E-mail: daniilivanov752@gmail.com  
  
class Stack:  
 def \_\_init\_\_(self):  
 *"""  
 Инициализация пустого стека.  
 """* self.items = []  
  
 def is\_empty(self):  
 *"""  
 Проверяет, пуст ли стек.  
 """* return len(self.items) == 0  
  
 def push(self, item):  
 *"""  
 Добавляет элемент на вершину стека.  
 :param item: Элемент, который нужно добавить.  
 """* self.items.append(item)  
  
 def pop(self):  
 *"""  
 Удаляет и возвращает элемент с вершины стека.  
 :return: Элемент с вершины стека.  
 :raises IndexError: Если стек пуст.  
 """* if not self.is\_empty():  
 return self.items.pop()  
 else:  
 raise IndexError("Стек пуст, невозможно выполнить операцию 'pop'.")  
  
 def peek(self):  
 *"""  
 Возвращает элемент с вершины стека без его удаления.  
 :return: Элемент с вершины стека.  
 :raises IndexError: Если стек пуст.  
 """* if not self.is\_empty():  
 return self.items[-1]  
 else:  
 raise IndexError("Стек пуст, невозможно выполнить операцию 'peek'.")  
  
 def size(self):  
 *"""  
 Возвращает количество элементов в стеке.  
 """* return len(self.items)  
  
 def \_\_str\_\_(self):  
 *"""  
 Возвращает строковое представление стека в удобном для человека виде.  
 """* return str(self.items)  
  
# Пример использования класса Stack:  
stack = Stack()  
stack.push(1)  
stack.push(2)  
stack.push(3)  
  
print("Стек:", stack) # Вывод: Стек: [1, 2, 3]  
  
top\_element = stack.pop()  
print("Извлеченный элемент:", top\_element) # Вывод: Извлеченный элемент: 3  
  
print("Размер стека:", stack.size()) # Вывод: Размер стека: 2

Стек: [1, 2, 3]

Извлеченный элемент: 3

Размер стека: 2

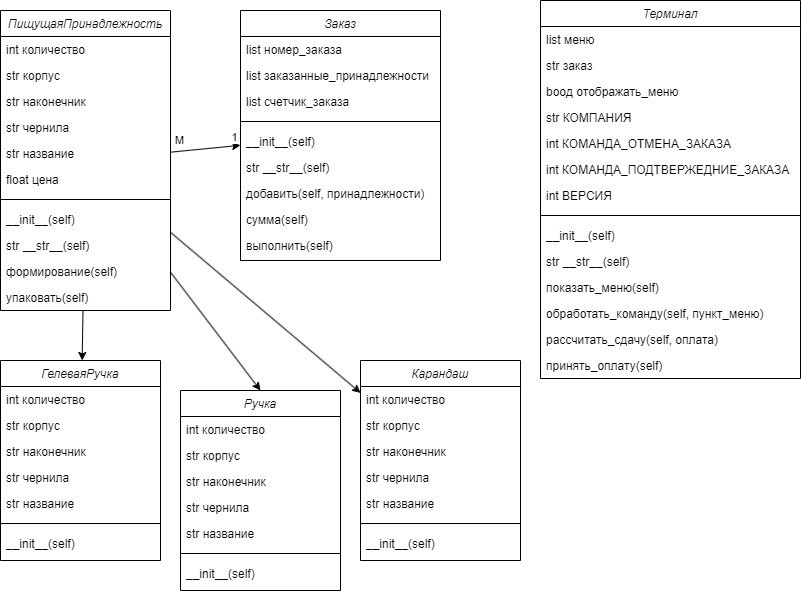


4.3.6. Иерархия классов

# Программирование на языке высокого уровня (Python).  
# Задание №4.3.6 Вариант 2  
#  
  
# Выполнил: Иванов Даниил Евгеньевич

# Группа: ПИН-б-о-22-1

# E-mail: daniilivanov752@gmail.com  
  
  
from ПищущаяПринадлежность import ПищущаяПринадлежность  
  
if \_\_name\_\_ == "\_\_main\_\_":  
  
 ПищущаяПринадлежность = ПищущаяПринадлежность()  
 print(ПищущаяПринадлежность)  
 while True:  
 ПищущаяПринадлежность.показать\_меню()  
 пункт\_меню = input()  
 ПищущаяПринадлежность.обработать\_команду(пункт\_меню)



Ссылка на полностью сделанные задания на github:

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