**Лабораторная работа №5**

**Тема:** объектно-ориентированное программирование в Python

**Цель:** приобрести навыки работы с классами в Python

**теоретическая часть**

**Класс** - это сложный пользовательский тип данных, состоящий из полей и методов. *Поля класса* - это переменные, объявленные внутри класса для хранения данных. *Методы класса* - это функции, объявленные для обработки полей класса, взаимодействия с основной программой и другими данными.

**Объекты** - это отдельные экземпляры класса, по сути переменные этого типа. Объявив класс с полями и методами, вы можете создавать сколько угодно объектов, каждый из которых будет содержать объявлен в классе набор полей и методов.

Классы объявляются с ключевым словом **class,** имя класса по стандартам Python записывается следующим образом: все слова вместе, каждое слово с большой буквы.

Для создания экземпляра класса вызывается как функция с круглыми скобками, возвращая новый объект, который вы можете сохранить в переменной.

Для методов класса можно указывать атрибуты.

**Атрибут** - это переменная, метод - это функция. Различия метода от функции в том, что у него есть первый параметр - **self.**

Классы, как и модули, скрывают внутреннее строение, оставляя на поверхности только внешний "интерфейс" для использования.

Это сочетание данных и функций внутри одной сущности, вместе с сокрытием внутреннего строения, называется *инкапсуляцией* и является главным принципом ООП.

При объявлении класса в скобках могут быть записаны (одно или несколько) имена уже существующих классов - это называется подражанием.В таком случае новый класс (который называется дочерним) наследует все поля и методы классов перечисленных в скобках (которые называются родительскими).

**Модель класса:**

class имя\_класса:

инструкция 1

....

инструкция N

**Создание объекта класса:**

обьект\_класу = имя\_класса ()

**Пример 1.** Программа с использованием класса. Вычисления среднего балла студента по трем предметам.

import math

class Student ():

def GPA (self, name, e1, e2, e3):

self. name = name

self.e1 = e1

self.e2 = e2

self.e3 = e3

print (self. name, '-', ((self.e1 + self.e2 + self.e3) / 3))

s1 = Student ()

s2 = Student ()

s1.GPA ( 'Dmytro', 5, 3, 4)

s2.GPA ( 'Olena', 5, 4, 5)

def main ():

return 0

if \_\_name\_\_ == '\_\_main\_\_':

main ()

Классы собирают в себе наборы данных (сменных) вместе с наборами функций, на них действуют. Цель состоит в том, чтобы достичь более модульного кода с помощью группировки переменных и функций, в небольшие узлы, легко модифицировать.

**Задание:**

1. Разработать класс "домашняя библиотека". Реализовать возможность работы с произвольным числом книг, поиска по книгам по нескольким параметрам (по автору, по году издания, по жанру и т.д.), добавление книг в библиотеку, удаления книг из нее, доступа к книге по номеру. Написать программу, которая будет демонстрировать все разработанные элементы класса.

2. Разработать класс для представления сведений об успеваемости студента. Объект класса должен содержать поля для сохранения имени студента и баллов, полученных им за выполнение лабораторных работ и индивидуального творческого задания.

Обеспечить следующие методы класса:

* конструктор, принимающий строку имя \_ студента и словарь, содержащий настройки курса в следующем формате:

1) максимально возможное количество баллов за сдачу индивидуального творческого задания;

2) максимально возможное количество баллов за сдачу одной лабораторной работы;

3) количество лабораторных работ в курсе;

4) доля баллов от максимума, которую необходимо набрать для получения экзамена автоматом.

* метод, помощью которого вносятся данные о количество попыток сдать лабораторную работу и оценка по последнюю попытку.
* метод, помощью которого вносятся данные о количество попыток сдать индивидуальное творческое Задание и оценка по последнюю попытку.
* метод, который возвращает кортеж (tuple), что содержит действительное число (сумму баллов студента по прохождения курса), и логическоезначение True или False в зависимости от того, достаточно этих баллов для получения оценки по экзамен автоматом.

3. Разработать класс, следует функциональность стандартного типа str и содержит 2 новых метода:

1) метод, который принимает 1 аргумент s и возвращает True или False в зависимости от того, содержит ли строка повторы последовательностей символов длиной от 3 символов.

2) метод, который возвращает True или False в зависимости от того, является ли строка палиндромом. Регистрами символов пренебрегать.Пустая строка считать палиндромом.

4. Разработать класс "колода карт", который будет включать закрыт массив элементов класса "карта". В карте будет храниться масть и номер.При создании экземпляра класса "колода карт", карты в колоде располагаются случайным образом. Обеспечить возможность вывода карты по телефону расположение в колоде, вывод всех карт, перемешивание колоды, выдачи одной карты из колоды, выдачи 6 карт из колоды. Написать программу, которая будет демонстрировать все разработанные элементы класса.

5. Разработать класс "англо-русский словарь", обеспечить возможность хранения нескольких вариантов перевода для каждого слова.Обеспечить возможность вывода всех вариантов перевода введенного английского слова.

6. Создать абстрактный класс "Транспортное средство". На его основе реализовать классы "Самолет", "Автомобиль" и "Корабль". Классы должны иметь возможность задавать и получать координаты и параметры средств передвижения (стоимость, скорость, год выпуска и т.д.) задать с помощью полей. Для самолета должна быть определена высота, для самолета и корабля - количество пассажиров, для корабля - порт приписки.Динамические характеристики задать с помощью методов.