**Лабораторная работа №1. Классы, пакеты, модули Python**

Цель работы: ознакомится с механизмом реализации классов, пакетов и модулей на Python. Унаследовать свой вариант класса от исходного базового класса, описывающего простейший случай данных реляционного вида (СУБД).

**Задание**

Составить программу, осуществляющую ввод и обработку массива элементов формируемый в класс. Размерность массива табличных данных задается пользователем, память под массив выделяется динамически. Разделить программу на два файла, реализацию класса поместить в отдельный пакет Python и импортировать ее в основной исполняемый модуль. Основные действия оформить в виде функций. Выполнить сортировку данных.

**1. Ход работы**

**1.1. Базовый класс данных**

Широко известные структуры данных используемые для описания сущностей на ПК формируются в виде двухмерных таблиц содержащие в себе различные типа данных и хранящиеся в файлах базах данных (БД). Можно описать простейший класс БД в Python следующим образом:

class MyDB:

def \_\_init\_\_(self, id):

self.id = id

def getTheId(self):

return self.id

def getMyData(self):

pass

def \_\_str\_\_(self):

return "Это представление данных класса BD"

Класс MyDB представляет из себя базовый функционал БД: установку у создаваемого объекта уникального номера (ID), методы – возвращения ID, получения данных и вывода строки состояния о объекте.

В процессе выполнения самостоятельного задания по лабораторной работе студенту требуется свой класс унаследовать от базового и реализовать свой вариант и функционал по обработке данных.

**1.2. Класс «книжная полка»**

На высокоуровневом языке программирования Python простейший класс книжной полки, содержащий данные: название книги, год издания и номер издания может быть представлен в следующем виде. Например,

class Books (MyDB):

def \_\_init\_\_(self, id, name = "", year = 0, edition = 0):

super().\_\_init\_\_(id)

self.name = name

self.year = year

self.edit = edition

def getMyData(self):

a = [self.name, self.year, self.edit]

return a

def setName(self, name):

self.name = name

def setYear(self, year):

self.year = year

def setEd(self, edition):

self.edit = edition

def \_\_str\_\_(self):

return "Книга с названием {0}, {1} года издания в {2} редакции".format(self.name, self.year, self.edit)

**1.3. Формирование пакета с классом**

Для многократного использования кода своего варианта класса описывающие разные данные удобно сформировать пакет Python. Проще говоря пакетом Python является файл с кодом расширения .py хранящийся в папке. Имя пакета – это имя папки. Имя файла – это модуль Python. Мы можем самостоятельно запускать модуль так и импортировать его. Поэтому предлагается в код модуля добавить следующий функционал по выводу информации о пакете.

if \_\_name\_\_ == "\_\_main\_\_":

d = dir(Books)

print("Класс Books содержит ", d)

one = Books(0)

print("Присутсвтуют следующие аттрибуты класса Books: ")

if hasattr(one, "name"):

print("Название книги")

if hasattr(one, "year"):

print("Год издания")

if hasattr(one, "edit"):

print("Номер издания")

else:

print("Импортировали модуль ", \_\_name\_\_)

**1.4. Импортирование модуля из пакета**

В нашем случае модуль имеет имя myModule и хранится в пакете lab3. Поэтому импортировать его в программу main.py мы можем следующим образом.

from lab3 import myModule as mm

Причем иерархия хранения файлов в ПК следующая:

main.py # Основная программа

lab3/ # Подпакет subA

myModule.py # Модуль

**1.6. Реализация наполнения структуры данных (объектов классов)**

Согласно своему варианту работы требуется создать свой класс и наполнить его данными для дальнейшей их обработки. Можно выполнить заполнение данными как в программе так и вводом данных с клавиатуры в консоль при запуске программы.

bookList = []

bookList.append(mm.Books(0, "Программирование на Python", 2009, 1))

bookList.append(mm.Books(1, "Программирование на С", 2001, 3))

bookList.append(mm.Books(2, "Программирование на Java", 2003, 5))

bookList.append(mm.Books(3, "Программирование на Ada", 1991, 1))

Для выполнения сортировки такого рода данных (списка из объектов класса) хранящего поля различного типа можно воспользоваться функцией sorted. Причем в параметр key передается имя функции, которая просто возвращает необходимое поле из класса:

def sortByYear(books):

return books.year

bookListSortedYear = sorted(bookList, key = sortByYear)

print("Отсортировали по году издания: ")

for b in bookListSortedYear:

print(b, end="; ")

В результате работы программы в консоль будет выведено следующее

Отсортировали по году издания:

Книга с названием Программирование на Ada, 1991 года издания в 1 редакции; Книга с названием Программирование на С, 2001 года издания в 3 редакции; Книга с названием Программирование на Java, 2003 года издания в 5 редакции; Книга с названием Программирование на Python, 2009 года издания в 1 редакции;

**2. Содержание отчета**

В состав отчета входит:

1. Титульный лист.
2. Задание
3. Код приложения из модуля myModule.py
4. Код приложения из основной программы main.py
5. Результат работы по варианту своего задания
6. Форматирование текста: номера страниц, верхний колонтитул

**3. Варианты заданий**

**Вариант 1**

Разработайте программу, которая, используя тип структура, формирует массив данных о сотрудниках отдела, содержащий следующую информацию: фамилию, год рождения и стаж работы. Затем определяет среди всех сотрудников, у которых стаж работы наибольший.

**Вариант 2**

Разработайте программу, которая формирует массив структур о студентах некоторой группы, содержащей следующую информацию: фамилию, оценки за последнюю сессию по трем предметам и размер стипендии. Фамилию и отметки программа должна вводить с клавиатуры, а размер стипендии считать, исходя из оценок: все "5" – повышенная (+25%), нет троек – обычная стипендия, есть тройки – стипендия 0. Базовый размер стипендии задать в программе константой. После чего программа должна вывести на экран результирующий массив.

**Вариант 3**

Разработайте программу, которая, используя тип структура, формирует массив данных о товарах на складе, содержащий следующую информацию: наименование товара, количество и цена. Затем сортирует полученный массив по цене и выводит отсортированный массив на экран, при этом рассчитывая стоимость товара (количество\*стоимость).

**Вариант 4**

Разработайте программу, которая, используя тип структура, формирует массив данных об объектах недвижимости для риэлтерского агентства. Элемент массива содержит следующую информацию: адрес, площадь, цена продавца. Отсортировать введенный массив по цене, вывести массив на экран, при этом рассчитывая цену 1 квадратного метра.

**Вариант 5**

Сведения о багаже каждого пассажира представляют собой элемент типа структура с тремя полями: ФИО пассажира, количество вещей и общий вес багажа в килограммах. Найти пассажиров, вес багажа которых превышает средний вес багажа всего массива пассажиров. Вывести на экран ФИО этих пассажиров.

**Вариант 6**

Описать, используя тип структура, данные на учеников (фамилия, улица, дом, квартира). Составить программу, определяющую, сколько учеников живет на заданной улице, списки учеников, живущих в заданном доме на этой улице.

**Вариант 7**

На олимпиаде по информатике на школьников заполнялись анкеты: фамилия, номер школы, класс, занятое место. Вывести списки учеников с количеством, занявших призовые места (1 место (кол-во) + список фамилий, 2 место …3 место…).

**Вариант 8**

В деканате хранится информация о зимней сессии на 1 курсе (фамилия, номер группы, оценка 1 по геометрии, оценка 2 по алгебре, оценка 3 по информатике). Составить программу, выводящую 1) фамилии студентов, имеющих задолженность хотя бы по одному предмету (оценка 2), 2) средний балл успеваемости на курсе.

**Вариант 9**

Разработайте программу, которая, используя тип структура, формирует массив данных о сотрудниках отдела, содержащий следующую информацию: ФИО, должность и общий стаж работы. Затем сортирует массив по стажу работы и выводит отсортированный массив на экран.

**Вариант 10**

Разработайте программу, которая, используя тип структура, формирует массив данных о товарах на складе, содержащий следующую информацию: наименование товара, цена. Затем ищет товар с минимальной ценой и товар с максимальной ценой, выводит результат.

**Вариант 11**

Разработайте программу, которая, используя тип структура, формирует массив данных о тур поездках, содержащий следующую информацию: страна, звездность отеля (2-5), цена. Затем ищет все туры в заданную страну и выводит результат с сортировкой по цене.

**Вариант 12**

Разработайте программу, которая, используя тип структура, формирует массив данных об авиабилетах (ФИО пассажира, возраст, вес багажа, стоимость билета). Стоимость билета рассчитывается, исходя из тарифа + доплата за перевес. Тариф, оплата за 1 кг багажа свыше норматива, норматив в кг задаются в программе константами. Детям от 2 до 12 лет скидка 30% от тарифа, детям до 2 лет – 100%. Подсчитать общий вес багажа на рейсе.

**Вариант 13**

Разработайте программу, которая, используя тип структура, формирует массив данных о зарплате (ФИО сотрудника, оклад, подоходный налог, премия, итого). Поля «подоходный налог» (12%) и «итого» рассчитываются автоматически. Подсчитать общую сумму выданной зарплаты.

**Вариант 14**

Используя тип структура, создать каталог публикаций, содержащий информацию – ФИО автора, название статьи, количество страниц, год издания. Выдать список публикаций не позднее заданного года издания (вводится пользователем).

**Вариант 15**

Пользователь вводит дату в формате строки «dd.mm.yy». Преобразовать строку в структуру с полями – день, название месяца, год.

**Вариант 16**

Массив из структур содержит данные о результатах соревнований по легкой атлетике: ФИО спортсмена, страна, год рождения, разряд, результат в секундах. Определить победителя и призеров (первые 3 места).

**Вариант 17**

Массив из структур содержит данные по среднемесячной температуре за год (месяц, температура). Вычислить среднегодовую температуру.

**Вариант 18**

Массив из структур содержит данные об автомобилях (марка, год выпуска, стоимость). Отсортировать массив по стоимости.