**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МО ЭВМ**

отчет

**по лабораторной работе №1**

**по дисциплине «Информатика»**

Тема: Парадигмы программирования

|  |  |  |
| --- | --- | --- |
| Студент гр. 3344 |  | Мурдасов М.К. |
| Преподаватель |  | Глазунов С.А. |

Санкт-Петербург

2024

## Цель работы

Получить базовые навыки работы с ООП на языке Python.

## Задание

**Базовый класс — печатное издание Edition:**

class Edition:

Поля объекта класса Edition:

* название (строка)
* цена (в руб., целое положительное число)
* возрастное ограничение (в годах, целое положительное число)
* стиль(значение может быть одной из строк: c (color), b (black))
* При создании экземпляра класса Edition необходимо убедиться, что переданные в конструктор параметры удовлетворяют требованиям, иначе выбросить исключение ValueError с текстом 'Invalid value'.

##### Книга - Book:

class Book: #Наследуется от класса Edition

Поля объекта класс Book:

* название (строка)
* цена (в руб., целое положительное число)
* возрастное ограничение (в годах, целое положительное число)
* стиль(значение может быть одной из строк: c (color), b (black))
* автор (фамилия, в виде строки)
* твердый переплет (значениями могут быть или True, или False)
* количество страниц (целое положительное число)
* При создании экземпляра класса Book необходимо убедиться, что переданные в конструктор параметры удовлетворяют требованиям, иначе выбросить исключение ValueError с текстом 'Invalid value'.

*В данном классе необходимо реализовать следующие методы:*

Метод \_\_str\_\_():

Преобразование к строке вида: Book: название <название>, цена <цена>, возрастное ограничение <возрастное ограничение>, стиль <стиль>, автор <автор>, твердый переплет <твердый переплет>, количество страниц <количество страниц>.

Метод \_\_eq\_\_():

Метод возвращает True, если два объекта класса равны и False иначе. Два объекта типа Book равны, если равны их название и автор.

##### Газета - Newspaper:

class Newspaper: #Наследуется от класса Edition

Поля объекта класс Newspaper:

* название (строка)
* цена (в руб., целое положительное число)
* возрастное ограничение (в годах, целое положительное число)
* стиль(значение может быть одной из строк: c (color), b (black))
* интернет издание (значениями могут быть или True, или False)
* страна (строка)
* периодичность (период выпуска газеты в днях, целое положительное число)
* При создании экземпляра класса Newspaper необходимо убедиться, что переданные в конструктор параметры удовлетворяют требованиям, иначе выбросить исключение ValueError с текстом 'Invalid value'.

*В данном классе необходимо реализовать следующие методы:*

Метод \_\_str\_\_():

Преобразование к строке вида: Newspaper: название <название>, цена <цена>, возрастное ограничение <возрастное ограничение>, стиль <стиль>, интернет издание <интернет издание>, страна <страна>, периодичность <периодичность>.

Метод \_\_eq\_\_():

Метод возвращает True, если два объекта класса равны и False иначе. Два объекта типа Newspaper равны, если равны их название и страна.

##### **Необходимо определить список list для работы с печатным изданием:** Книги:

class BookList – список книг - наследуется от класса list.

Конструктор:

1. Вызвать конструктор базового класса.
2. Передать в конструктор строку name и присвоить её полю name созданного объекта

*Необходимо реализовать следующие методы:*

Метод append(p\_object): Переопределение метода append() списка. В случае, если p\_object - книга, элемент добавляется в список, иначе выбрасывается исключение TypeError с текстом: Invalid type <тип\_объекта p\_object> (результат вызова функции type)

Метод total\_pages(): Метод возвращает сумму всех страниц всех имеющихся книг.

Метод print\_count(): Вывести количество книг.

##### Газеты:

class NewspaperList – список газет - наследуется от класса list.

Конструктор:

1. Вызвать конструктор базового класса.
2. Передать в конструктор строку name и присвоить её полю name созданного объекта

*Необходимо реализовать следующие методы:*

Метод extend(iterable): Переопределение метода extend() списка. В случае, если элемент iterable - объект класса Newspaper, этот элемент добавляется в список, иначе не добавляется.

Метод print\_age(): Вывести самое низкое возрастное ограничение среди всех газет.

Метод print\_total\_price(): Посчитать и вывести общую цену всех газет.

В отчете укажите:

1. Изображение иерархии описанных вами классов.
2. Методы, которые вы переопределили (в том числе методы класса object).
3. В каких случаях будут использованы методы \_\_str**\_\_**() и \_\_eq**\_\_**().
4. Будут ли работать переопределенные методы класса list для BookList и NewspaperList? Объясните почему и приведите примеры.

## Выполнение работы

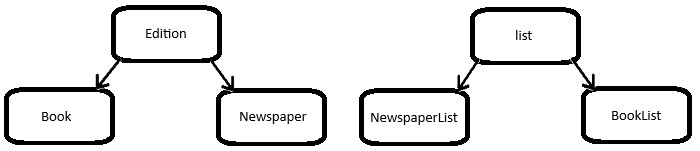


Рисунок 1 – Изображение иерархии классов

1. Методы, унаследованные у *Edition*:

*\_\_init\_\_()* – Принимает параметры издания и проверяет на корректность

*\_\_str\_\_()* – Преобразовывает данные об издании в строку и возвращает ее

*\_\_eq\_\_()* – Сравнивает два объекта одного класса, проверяет издания на идентичность

1. Методы, унаследованные у *list*:

*\_\_init\_\_()* – Принимает параметры издания и проверяет на корректность

*append()* – Добавляет объект в список, если он является книгой, в противном случае выводит ошибку

*total\_pages()* – Возвращает сумму всех страниц всех имеющихся книг

*print\_count()* – Выводит количество книг

*extend()* – Добавляет в список только элементы класса *Newspaper*

*print\_age()* – Выводит минимальное возрастное ограничение среди всех газет

*print\_total\_price()* – Выводит суммарную стоимость всех газет

1. Метод *\_\_str\_\_()* будет применяться при вызове *str(<издание>)*

Метод *\_\_eq\_\_()* будет применяться при сравнении изданий одного класса.

1. Переопределенные методы класса *list* будут работать для *BookList* и *NewspaperList*, т.к. к ним лишь были добавлены условия, а в остальном функции методов остались прежними.

**Тестирование**

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Входные данные | Выходные данные | Комментарии |
|  | print('----------------------BOOK----------------------')  book = Book("ахахаха", 999, 1, "b", "Я", True, 1889)  book2 = Book("ахахаха", 111, 56, "c", "Я", False, 1945)  book3 = Book("хох", 111, 9, "c", "НЕ Я", False, 1945)  book\_str = Book.\_\_str\_\_(book)  book2\_str = Book.\_\_str\_\_(book2)  print(f'book1 - {book\_str}\nbook2 - {book2\_str}\nbook == book2: {book == book2}\nbook == book3: {book == book3}')  print('----------------------NEWSPAPER----------------------')  newspaper = Newspaper("Ниче такая газета", 1, 8, "c", True, "Russia", 7)  newspaper2 = Newspaper("Ниче такая газета", 500, 99, "b", True, "Russia", 7)  newspaper3 = Newspaper("Че такая газета", 500, 99, "b", True, "Не Russia", 7)  newspaper\_str = Newspaper.\_\_str\_\_(newspaper)  newspaper2\_str = Newspaper.\_\_str\_\_(newspaper2)  print(f'newspaper1 - {newspaper\_str}\nnewspaper2 - {newspaper2\_str}\nnewspaper == newspaper2: {newspaper == newspaper2}\nnewspaper == newspaper3: {newspaper == newspaper3}')  print('----------------------BOOKLIST----------------------')  booklist = BookList(Book)  booklist.append(book)  booklist.append(book2)  booklist.append(book3)  print(f'total\_pages(): {booklist.total\_pages()}\nprint\_count():')  booklist.print\_count()  print('----------------------NEWSPAPERLIST----------------------')  news\_list = NewspaperList(Newspaper)  news\_list.extend([newspaper, newspaper2, book, book2, newspaper3])  print('Added elements:', news\_list)  print('print\_age:')  news\_list.print\_age()  print('print\_total\_price:')  news\_list.print\_total\_price() | ----------------------BOOK----------------------  book1 - Book: название ахахаха, цена 999, возрастное ограничение 1, стиль b, автор Я, твердый переплет True, количество страниц 1889.  book2 - Book: название ахахаха, цена 111, возрастное ограничение 56, стиль c, автор Я, твердый переплет False, количество страниц 1945.  book == book2: True  book == book3: False  ----------------------NEWSPAPER----------------------  newspaper1 - Newspaper: название Ниче такая газета, цена 1, возрастное ограничение 8, стиль c, интернет издание True, страна Russia, периодичность 7.  newspaper2 - Newspaper: название Ниче такая газета, цена 500, возрастное ограничение 99, стиль b, интернет издание True, страна Russia, периодичность 7.  newspaper == newspaper2: True  newspaper == newspaper3: False  ----------------------BOOKLIST----------------------  total\_pages: 5779  print\_count():  3  ----------------------NEWSPAPERLIST----------------------  Added elements: [<\_\_main\_\_.Newspaper object at 0x00000176174ED690>, <\_\_main\_\_.Newspaper object at 0x00000176174EF790>, <\_\_main\_\_.Newspaper object at 0x00000176174EF7D0>]  print\_age:  8  print\_total\_price:  1001 | Корректно |

## Выводы

Были получены базовые навыки работы с ООП. С использованием ООП была написана программа для работы с информацией о печатных изданиях. Также были изучены наследование классов и переопределение методов.

# Приложение А Исходный код программы

Название файла: Murdasov\_Mikhail\_lb1.py

class Edition:

def \_\_init\_\_(self, name, price, age\_limit, style):

if not(isinstance(price, int) and price > 0 and isinstance(age\_limit, int) and age\_limit > 0 and (style == "c" or style == "b") and isinstance(name, str)):

raise ValueError("Invalid value")

self.name = name

self.price = price

self.age\_limit = age\_limit

self.style = style

class Book(Edition):

def \_\_init\_\_(self, name, price, age\_limit, style, author, hardcover, pages):

super().\_\_init\_\_(name, price, age\_limit, style)

if not(isinstance(author, str) and isinstance(hardcover, bool) and isinstance(pages, int) and pages > 0):

raise ValueError("Invalid value")

self.author = author

self.hardcover = hardcover

self.pages = pages

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

return f"Book: название {self.name}, цена {self.price}, возрастное ограничение {self.age\_limit}, стиль {self.style}, автор {self.author}, твердый переплет {self.hardcover}, количество страниц {self.pages}."

def \_\_eq\_\_(self, other):

return self.name == other.name and self.author == other.author

class Newspaper(Edition):

def \_\_init\_\_(self, name, price, age\_limit, style, online\_edition, country, frequency):

super().\_\_init\_\_(name, price, age\_limit, style)

if not(isinstance(online\_edition, bool) and isinstance(country, str) and isinstance(frequency, int) and frequency > 0):

raise ValueError("Invalid value")

self.online\_edition = online\_edition

self.country = country

self.frequency = frequency

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

return f"Newspaper: название {self.name}, цена {self.price}, возрастное ограничение {self.age\_limit}, стиль {self.style}, интернет издание {self.online\_edition}, страна {self.country}, периодичность {self.frequency}."

def \_\_eq\_\_(self, other):

return self.name == other.name and self.country == other.country

class BookList(list):

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

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

self.name = name

def append(self, object):

if isinstance(object, Book):

super().append(object)

else:

raise TypeError(f"Invalid type <тип\_объекта {type(object)}>")

def total\_pages(self):

return sum([x.pages for x in self])

def print\_count(self):

print(len(self))

class NewspaperList(list):

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

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

self.name = name

def extend(self, iterable):

super().extend([x for x in iterable if isinstance(x, Newspaper)])

def print\_age(self):

print(min([x.age\_limit for x in self]))

def print\_total\_price(self):

print(sum([x.price for x in self]))