ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ РЕСПУБЛИКИ ХАКАСИЯ

«ХАКАССКИЙ ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ»

Специальность 09.02.07

«Информационные системы и программирование»

Практическое занятие №3

тема: создание классов

Руководитель:

\_\_\_\_\_\_\_\_ А.А. Отыргашева

(подпись)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(оценка, дата)

Выполнил:

Студент группы ИС(ПРО)-31

\_\_\_\_\_\_\_\_ Д. А. Коробкова

(подпись)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(дата)

Абакан 2025 г.

ЗАДАНИЕ №1

class Book: #определение нового класса

def \_\_init\_\_ (self, author, pages, year): #метод, принимающий параметры класса

self.author = author #установка атрибута “автор”, который равен значению “автор”

self.pages = pages #установка атрибута “страницы”, который равен значению “страницы”

self.year = year #установка атрибута “год”, который равен значению “год”

def get\_author(self): #метод для возвращения значения атрибута “автор”

return self.author #возвращение значения атрибута “автор”

def set\_author(self, author): #метод для установки нового значения в атрибут “автор”

self.author = author # присваивание значения параметра автор атрибуту self.author

def get\_pages(self): #метод для возвращения значения атрибута “страницы”

return self.pages #возвращение значения атрибута “страницы”

def set\_pages(self, pages): #метод для установки нового значения в атрибут “страницы”

self.pages = pages # присваивание значения параметра страницы атрибуту self.pages

def get\_year(self): #метод для возвращения значения атрибута “год”

return self.year #возвращение значения атрибута “год”

def set\_year(self, year): #метод для установки нового значения в атрибут “год”

self.year = year # присваивание значения параметра год атрибуту self.year

author = input("Введите автора книги: ") #запрос ввода ФИО автора у пользователя

pages = int(input("Введите количество страниц в книге: ")) #запрос ввода количества страниц у пользователя

year = int(input("Введите год написания книги: ")) #запрос ввода года написания книги у пользователя

knizhka = Book(author, pages, year) #создание нового объекта класса

print() # данный print выводит пустую строку и необходим для более комфортного глазу чтения

print(f"Автор книги: {knizhka.author}, в ней {knizhka.pages} страниц(-ы), а написана она в {knizhka.year} году") # вывод всех данных в тексте

ЗАДАНИЕ №2

class Book: # определение нового класса  
 def \_\_init\_\_(self, author, pages, year, title): # метод, принимающий параметры класса  
 self.author = author # установка атрибута «автор», который равен значению «автор»  
 self.pages = pages # установка атрибута «страницы», который равен значению «страницы»  
 self.year = year # установка атрибута «год», который равен значению «год»  
 self.title = title # установка атрибута «название», который равен значению «название»  
  
 def \_\_str\_\_(self): # определение функции для вывода сообщения  
 return f"Автор книжки: {self.author}, назвается '{self.title}', количество страничек: {self.pages}, написана в {self.year} г." # функция, которая будет возвращать сообщение, в котором будут содержать данные из кода  
  
author = input("Введите имя автора: ") # запрос ввода данных «Имя автора» от пользователя   
title = input("Введите название книги: ")# запрос ввода данных «Название книги» от пользователя  
pages = int(input("Введите количество страниц: "))# запрос ввода данных «Количество страниц» от пользователя  
year = int(input("Введите год издания: "))# запрос ввода данных «Год издания» от пользователя  
  
if not author.isalpha(): # проверка данных переданных в переменную author на то, что строка состоит только из букв  
 print("Автор должен содержать только буквы") # вывод в консоль сообщения  
elif year > 2025: # проверка, если переменная year больше 2025, то выполняется следующий блок кода  
 print("Год не может быть больше 2025")# вывод в консоль сообщения  
elif not 5 <= pages <= 2000: # проверка, если количество страниц не находится в диапазоне от 5 до 2000 включительно, то выполняется следующий блок кода

print("Страниц должно быть от 5 до 2000")# вывод в консоль сообщения  
elif len(title) > 100: # проверка, если длина строки в переменной title больше 100, то выполняется следующий блок кода  
 print("Название слишком длинное")# вывод в консоли сообщения  
else: # данный блок кода выполняется, если ни одно из предыдущих условий не было истинным.  
 book = Book(author, title, pages, year) # создание нового объекта Book, используя переданные параметры: author, title, pages, и year  
 print(book) # вывод в консоль информации об объекте book

ЗАКЛЮЧЕНИЕ

В ходе выполнения работы были успешно реализованы два задания, касающиеся создания и работы с классами в языке программирования Python.

Задание 1 заключалось в создании класса с заданными полями, обеспечивающими доступ к значениям этих полей. В результате была разработана программа, демонстрирующая создание двух экземпляров класса и вывод значений их свойств.

Задание 2 потребовало более глубокого подхода к проектированию класса с учетом ограничений предметной области. В рамках этого задания был создан класс, который включает свойства и методы, обеспечивающие контроль доступа к полям с учетом заданных ограничений. Конструктор класса был реализован таким образом, чтобы гарантировать корректное создание объектов, соответствующих этим ограничениям. Программа, демонстрирующая работу с двумя экземплярами данного класса, подтвердила, что ограничения сохраняются при любых действиях с объектами, что является важным аспектом при разработке надежного программного обеспечения.

Таким образом, выполненные задания не только углубили понимание принципов объектно-ориентированного программирования, но и подчеркнули значимость правильной архитектуры классов для обеспечения целостности и надежности данных. Полученные знания и навыки будут полезны в дальнейшей практике программирования и разработке сложных систем.