**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ   
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ТЕЛЕКОММУНИКАЦИЙ И ИНФОРМАТИКИ»**

**ХАБАРОВСКИЙ ИНСТИТУТ ИНФОКОММУНИКАЦИЙ (ФИЛИАЛ)**

**СРЕДНЕЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАНИЕ**

КУРСОВОЙ ПРОЕКТ

по МДК 01.02 Прикладное программирование

на тему

Программа формирования списков «Библиотека»

Выполнил: А.В. Медведев

Проверил: А.Н. Поляков

Группа: ПКС-320

2022

СОДЕРЖАНИЕ

|  |  |
| --- | --- |
| Введение | 3 |
| 1. Постановка задача | 5 |
| 1.1 Описание задачи | 5 |
|  |  |
|  |  |

ВВЕДЕНИЕ

Современные крупные предприятия ежедневно работают с данными. Данные — поддающееся многократной интерпретации представление информации в формализованном виде, пригодном для передачи, связи или обработки. Информация о сотрудниках корпорации, продуктах гипермаркета, медицинском центре может быть организована в базы данных. Обычно это делается для обеспечения одновременного доступа к данным множества пользователей, нередко расположенных достаточно далеко друг от друга. В них тем или иным путем решаются специфические проблемы параллельных процессов, целостности (правильности) и безопасности данных. Целью данной курсовой работы является разработка программы формирования списков библиотеки, которая должна решать задачу регистрации, обработки и хранения информации о читателях. База данных для читателей должна содержать всю необходимую личную информацию, а также адрес проживания.

Курсовой проект будет выполняться в высокоуровневом языке программирования Python при помощи интегрированной среды разработки PyCharm.

Чтобы перейти к постановке задачи проекта следует дать определение использованной парадигмы программирования и основных её концепций.

Объектно-ориентированное программирование (ООП) — это подход, при котором программа рассматривается как набор объектов, взаимодействующих друг с другом. Объектно-ориентированное программирование используется, с целью:

1. структурировать информацию и не допускать путаницы;
2. точно определять взаимодействие одних элементов с другими;
3. повышать управляемость программы;
4. быстрее масштабировать код под различные задачи;
5. лучше понимать написанное;
6. эффективнее поддерживать готовые программы;
7. внедрять изменения без необходимости переписывать весь код.

Объектно-ориентированное программирование основано на принципах:

1. абстракции – способ выделить набор наиболее важных атрибутов и методов и исключить незначимые;
2. инкапсуляции – объединение функций и данных в рамках одной структуры, внутреннее состояние которой (данные) скрыто от внешнего мира;
3. наследование – свойство системы, позволяющее описать новый класс на основе уже существующего с частично или полностью заимствующейся функциональностью.
4. полиморфизм – свойство системы использовать объекты с одинаковым интерфейсом без информации о типе и внутренней структуре объекта.
5. ПОСТАНОВКА ЗАДАЧИ

1.1 Описание задачи

Программа формирования списков “Библиотека” обрабатывает исходные данные и выводит требуемые списки по запросу пользователя. Программа исполняется в консоли операционной системы. Данные о читателях библиотеки хранятся в базе данных. Работа с программой осуществляется посредством взаимодействия пользователя с базой данных при помощи языка структурированных запросов. На выбор пользователю даётся набор функций, для взаимодействия с программой:

1. добавить запись;
2. удалить запись;
3. обновить запись;
4. вывести списки;
5. помощь;
6. экспорт базы данных в формат JSON.

Выходная информация о читателях сортируется по номеру читательского билета.

1.2 Описание состава входной и выходной информации

Входной информацией являются сведения о читателях: номер читательского билета, фамилия и инициалы читателя, год рождения читателя, возраст, образование (высшее, среднее, школьное), пол, адрес.

Выходной информацией является список, сформированный по требованию пользователя:

1. по заданному пользователем диапазону возраста и образованию;
2. по полу;
3. по образованию, отсортированном по году рождения читателя;
4. по заданному пользователем диапазону номера читательского билета
5. по заданной пользователем букве фамилии, отсортированной по возрасту.

1.3 Задачи курсового проекта

Задачей курсового проекта является формирование программы для работы с данными о читателях библиотеки. Необходимо:

1. разработать черновой прототип программы;
2. составить базу данных читателей и определить необходимые поля для описания их;
3. разработать программу для взаимодействия библиотекаря с базой данных читателей;
4. составить и устранить список возможных ошибок, связанных с работой в программе;

1.4 Ограничения на входные данные и методы их контроля

Пользователь вводит данные в соответствии с их типом данных. К целочисленному типу данных относится номер читательского билета, год рождения и возраст читателя. К строковому типу данных относится фамилия и инициалы, адрес проживания и уровень образования читателя.

В программе предусмотрены проверки на соответствие вводимых данных их типу, а также на область допустимых значений для данного поля.

def surname\_np(self, value):

while True:

try:

k = input("Введите фамилию, имя и отчество\t")

except (TypeError, ValueError) as err:

print("Введите допустимую запись ФИО")

else:

try:

surname, name, patronymic = k.split()

if surname.isdigit() or name.isdigit() or patronymic.isdigit():

print("Введите правильное ФИО\n\tНапример: Бузова Ольга Петровна")

else:

lower = surname.lower()

self.\_\_surname\_np = f"{lower.replace(lower[0], lower[0].upper(), 1)} {name[0].upper()}.{patronymic[0].upper()}.".format( \*\*vars())

break

except (TypeError, ValueError) as err:

print("Введите правильное ФИО\n\tНапример: Бузова Ольга Петровна")

2. ФУНКЦИОНАЛЬНАЯ СХЕМА ПРОГРАММЫ

На рисунке 2.1 представлена функциональная схема программы.

 Рисунок 2.1 - Функциональная схема программы

3. БЛОК-СХЕМА ПРОГРАММЫ