Работу выполнили

студенты группы ИС-33

Клименко О.Ю.

Сапонджян Х.С.

**Практическая работа №16.**

**Библиотека**

Описание: Проект "Библиотека" представляет собой программу для управления библиотечными ресурсами, включая книги, клиентов и сотрудников. В рамках проекта реализованы следующие классы и интерфейсы:

**Класс Main:**

Это основной класс программы, который содержит метод main для запуска приложения. Внутри этого класса создаются экземпляры классов для книг, клиентов и сотрудников. Он демонстрирует использование основной функциональности библиотеки, а также отображение информации о книгах и авторах.

**Класс Library:**

Предназначен для представления информации о библиотеке. Внутри класса объявлены приватные переменные title, address, year, name и статическая переменная online.

Общая цель класса Library состоит в предоставлении базовой информации о библиотеке, такой как название, адрес, год основания и онлайн-статус, а также

обеспечении функциональности для отображения этой информации.

**Класс Book**:

Данный класс представляет книгу с определенными атрибутами, такими как автор, год выпуска, название, язык.

**Класс Person:**

представляет человека с основными атрибутами имени и фамилии.

Этот класс служит в качестве базового класса для более конкретных типов людей, которые могут быть представлены в проекте. Он обеспечивает базовую функциональность для получения и установки имени и фамилии человека.

**Класс Client:**

предназначен для представления клиентов библиотеки. Он хранит информацию о каждом клиенте, включая личные данные из объекта Person, информацию о задолженностях и уникальный идентификатор клиента.

**Класс Writer:**

представляет писателя и содержит информацию о его дате рождения, дате смерти и персональные данные.

**Интерфейс Printable:**

Путем реализации интерфейса Printable классы могут обеспечить удобный способ вывода информации, что упрощает разработку и сопровождение программы.

**Интерфейс Salary:**

Интерфейс Salary с методом Salarys используется для определения действий, связанных с зарплатой или финансовыми аспектами, которые могут быть реализованы различными классами. Использование модификатора final в объявлении метода Salarys гарантирует, что этот метод не может быть переопределен в подклассах

**Класс MultimediaMaterial:**

Этот класс предназначен для представления мультимедийных материалов в библиотеке. У класса есть несколько дополнительных атрибутов, таких как format (формат), duration (продолжительность) и type (тип).

**Класс Event:**

предназначен для представления событий, связанных с библиотекой. У класса есть три дополнительных атрибута: Eventname (название события), Descriptionn (описание) и dateAndTime (дата и время проведения).

**Класс ReadingClub:**

предназначен для представления читательского клуба, связанного с библиотекой. У класса есть три дополнительных атрибута: peoplelist (список участников), rules (правила) и resources (ресурсы).

**Класс Worker:**

реализует интерфейс Salary. Класс Worker предназначен для представления информации о сотрудниках библиотеки. У класса есть три атрибута: isMan (пол), name (имя) и id (идентификационный номер сотрудника).

Метод Salarys реализует интерфейс Salary и позволяет рассчитать зарплату сотрудника в зависимости от его должности. Метод запрашивает у пользователя ввод должности сотрудника и, на основе введенной должности, выводит информацию о зарплате.

**Театр.**

Описание: проект “Театр” представляет собой программу для управления ресурсами театра, включая комнаты, сотрудников и билеты. В рамках проекта реализованы следующие классы и интерфейсы:

**Класс Main:**

Это основной класс программы, который содержит метод main для запуска приложения. Внутри этого класса создаются экземпляры классов для залов, сотрудников и билетов. Он демонстрирует использование основной функциональности театра, а также отображение информации о сотрудниках и билетах.

**Класс Theater:**

Предназначен для представления информации о театра. Внутри класса объявлены приватные переменные title, address, year, name, canteen и type. Общая цель класса Theater состоит в предоставлении базовой информации о театре, такой как название, адрес, год основания, имя основателя, информация о реализации буфета внутри театра и вид самого театра, а также

обеспечении функциональности для отображения этой информации.

**Класс MainHall:**

Класс представляет собой один из залов внутри театра и содержит информацию о количествах комнат внутри зала, скамеек и есть ли хотя бы одно зеркало внутри комнаты.

**Класс Amphitheater:**

В классе обозначаются такие свойства как: диаметр полукруга афитиатра, материал на сиденьях(скамейках) и слышимость(хорошая или плохая).

**Класс Parterre:**

Данный класс имеет следующие свойства: количество мест, надобность в пенсне и есть ли яма с оркестром.

**Класс Mezzanine:**

В классе описывается на каком ярусе расположен бельэтаж, есть ли официанты на этаже и количество входов/выходов.

**Класс Balcony:**

В классе свойства высоты балкона, есть ли отдельный туалет и длина балкона.

**Класс Toilet:**

Данный класс имеет целых 4 свойства: количество унитазов, есть ли писсуары и их количество, есть ли разделение на женские и мужские санузлы.

**Класс Wardrobe:**

Предназначен для указания количества вешалок, гардеробщиков и времени работы гардероба.

**Класс Ticket:**

Класс имеет свойства билета, в котором написано: Название предстваления, дата его проведения и цена.  
В данном классе реализуется метод static, применяемый для указания постоянной цена на все билеты(1499,99 руб.).

Также в этом классе есть функция addTicket(), с помощью которой пользователь может ввести количество нужных ему билетов и узнать стоимость.

**Класс Employee:**

Класс сотрудников, в котором прописывается ID сотрудника, ФИО и должность.

**Класс EmployeeList:**

Класс, благодаря мы можем записывать данные о сотрудниках в лист.

Также в этом классе реализуется функция getEmplyeeById(), в которой мы может находить ранее занесенного сотрудника по его ID и так же возвращать null, если сотрудник не найдет.

Данный класс использует интерфейс **Salary**, благодаря которому мы можем узнавать зарплату определенных сотрудников.

Также стоит отметить, что классы **Employee** и **EmployeeList** - не являются наследниками от класса **Theater**, поэтому они выводят только информацию, которая занесена в конструктор каждого класса.