МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

(ФГБОУ ВО «ВятГУ»)

Институт математики и информационных систем

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Отчёт

Лабораторная работа №1 по дисциплине

«Объектно-ориентированное программирование»

Выполнил студент группы ИВТб-2305-05-00 \_\_\_\_\_\_\_\_ /Огородникова Д.А.

Проверил зав. кафедры ЭВМ \_\_\_\_\_\_\_\_ /Долженкова М.Л.

Киров 2025

**Цель работы:** создать иерархию классов, состоящую не менее чем из одного родительского и двух дочерних классов. В каждом классе определить не менее двух член-данных, не менее двух собственных, а для дочерних не менее двух унаследованных и двух перекрытых член-функций.

**Задание:** разработать консольное приложение, демонстрирующее принципы инкапсуляции, наследования и полиморфизма.

1. **Описание предметной области**

Предметная область «Фильмы». Были выделены следующие классы:

1. Фильмы — базовый класс, содержащий общие характеристики и методы для всех типов фильмов.
2. Развлекательный фильм — наследует базовый класс и добавляет специфические характеристики и методы для развлекательных фильмов.
3. Научный фильм — наследует базовый класс и добавляет специфические характеристики и методы для научных фильмов.
4. **Схема структуры иерархии классов**

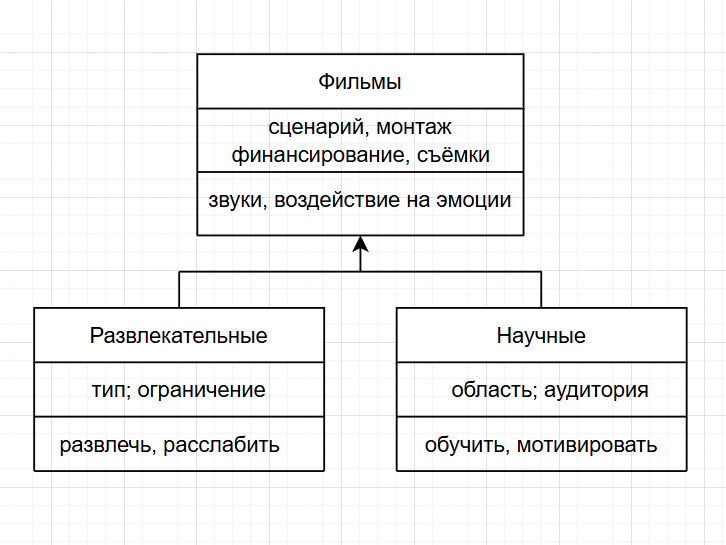


Рисунок 1 — Диаграмма классов

1. Фильмы — родительский класс, содержит общие данные и методы для всех фильмов.
2. Развлекательные и научные — дочерние классы, наследуют свойства и методы родительского класса, а также добавляют специфические для каждого типа фильма.
3. **Особенности реализации**

В программе были реализованы следующие принципы объектно-ориентированного программирования:

1. Инкапсуляция:

В базовом классе Film и его дочерних классах (EntertainmentFilm, ScienceFilm) данные (scenario, montage, type, ageRestriction, field, audience) объявлены как protected или private, что ограничивает прямой доступ к ним извне.

Для работы с этими данными используются публичные методы-геттеры и сеттеры: getScenario(), setScenario(), getMontage(), setMontage(), getType(), setType(), getAgeRestriction(), setAgeRestriction(), getField(), setField(), getAudience(), setAudience().

Это обеспечивает контроль над изменением данных и защищает внутреннее состояние объектов.

1. Наследование:

Классы EntertainmentFilm и ScienceFilm наследуют базовый класс Film, что позволяет использовать общие поля и методы (scenario, montage, financing(), filming()).

В дочерних классах добавлены специфические методы:

• в EntertainmentFilm: entertain(), relax();

• в ScienceFilm: educate(), motivate().

1. Полиморфизм:

В программе используется виртуальный метод sounds() и emotions(), который переопределен в дочерних классах. Это позволяет вызывать метод через указатель на базовый класс, при этом будет выполнен метод соответствующего дочернего класса. В меню (switch-case) реализована логика работы с объектами через указатель на базовый класс Film, что демонстрирует принцип полиморфизма.

**Вывод:** в ходе выполнения работы были успешно реализованы принципы объектно-ориентированного программирования:

1. Инкапсуляция: данные классов защищены от прямого доступа, что обеспечивает контроль над изменением данных.
2. Наследование: дочерние классы расширяют функциональность базового класса, что позволяет использовать общие методы и данные.
3. Полиморфизм: переопределение методов в дочерних классах позволяет вызывать методы через указатель на базовый класс, что демонстрирует гибкость и расширяемость программы.

Работа с объектами осуществляется через переменные родительского типа, что соответствует требованиям задания.