

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Дата сдачи на проверку:  
« \_\_\_\_ » \_\_\_\_\_ 2025 г.

Проверено:  
« \_\_\_\_ » \_\_\_\_\_ 2025 г.

**ООП в C++**  
Отчет по лабораторной работе № 1  
по дисциплине  
«Объектно-ориентированное программирование»

Выполнил студент гр. ИВТб-2303-05-00 \_\_\_\_\_ /Крупница Р.А./  
(Подпись)

Руководитель \_\_\_\_\_ /Шмакова Н.А./  
(Подпись)

Работа защищена « \_\_\_\_ » \_\_\_\_\_ 2025 г.

Киров 2025

## 1 Цель работы

Закрепить навыки объектно-ориентированного программирования, научиться использовать наследование, полиморфизм и виртуальные функции на примере иерархии классов, описывающих музыкальные инструменты.

## 2 Задание

Разработать и реализовать на языке C++ консольное приложение, моделирующее взаимодействие с различными музыкальными инструментами. Предусмотреть базовый класс и производные классы (гитара, пианино), каждый из которых реализует специфичное поведение.

## 3 Описание реализации

В рамках лабораторной работы был создан базовый класс `MusicalInstrument`, включающий основные характеристики музыкальных инструментов — имя и количество струн/клавиш. В классе реализованы виртуальные методы для описания, настройки, игры и уникальной особенности инструмента.

От базового класса унаследованы:

- **Guitar** — содержит дополнительную информацию о типе гитары (акустическая или электрическая) и наличие звукоснимателя;
- **Piano** — реализует свойства пианино или рояля, а также количество педалей;

В главной функции реализовано меню взаимодействия: добавление, удаление и выбор инструмента с дальнейшим вызовом методов.

## 4 Описание свойств и методов классов

### Базовый класс `MusicalInstrument`

#### Свойства:

- `name` — имя инструмента;
- `stringsCount` — количество струн инструмента.

#### Методы:

- `play()` — выводит сообщение о том, что инструмент играет.
- `tune()` — имитирует настройку инструмента.
- `describe()` — выводит общее описание инструмента (имя и количество струн/клавиш).
- `specialFeature()` — демонстрирует уникальную особенность инструмента.
- `setName(name)` — устанавливает новое имя инструмента.
- `getName()` — возвращает текущее имя инструмента.
- `setStringsCount(count)` — задаёт количество струн/клавиш.
- `getStringsCount()` — возвращает текущее количество струн/клавиш.

### Производный класс `Guitar`

#### Свойства:

- `guitarType` — тип гитары (акустическая/электрическая);
- `hasPickup` — наличие звукоснимателя.

#### Методы:

- `strum()` — имитирует перебор струн гитары.
- `changeStrings()` — показывает процесс замены струн на гитаре.
- `describe()` — переопределяет базовый метод, добавляя информацию о типе гитары и наличии звукоснимателя.
- `specialFeature()` — переопределяет базовый метод, уточняя, что гитара способна играть как аккорды, так и мелодии.

## Производный класс Piano

### Свойства:

- `pedalCount` — количество педалей;
- `isGrand` — является ли пианино роялем (у рояля струны и механика расположены горизонтально. Молоточки возвращаются под действием силы тяжести, что делает игру более чувствительной и быстрой).

### Методы:

- `pressPedal()` — имитирует нажатие педали.
- `cleanKeys()` — отображает процесс очистки клавиш.
- `describe()` — переопределяет базовый метод, выводит тип (рояль или пианино), количество клавиш и педалей.
- `specialFeature()` — переопределяет базовый метод, поясняет возможность одновременного исполнения мелодии и аккомпанемента.

## 5 Диаграмма классов

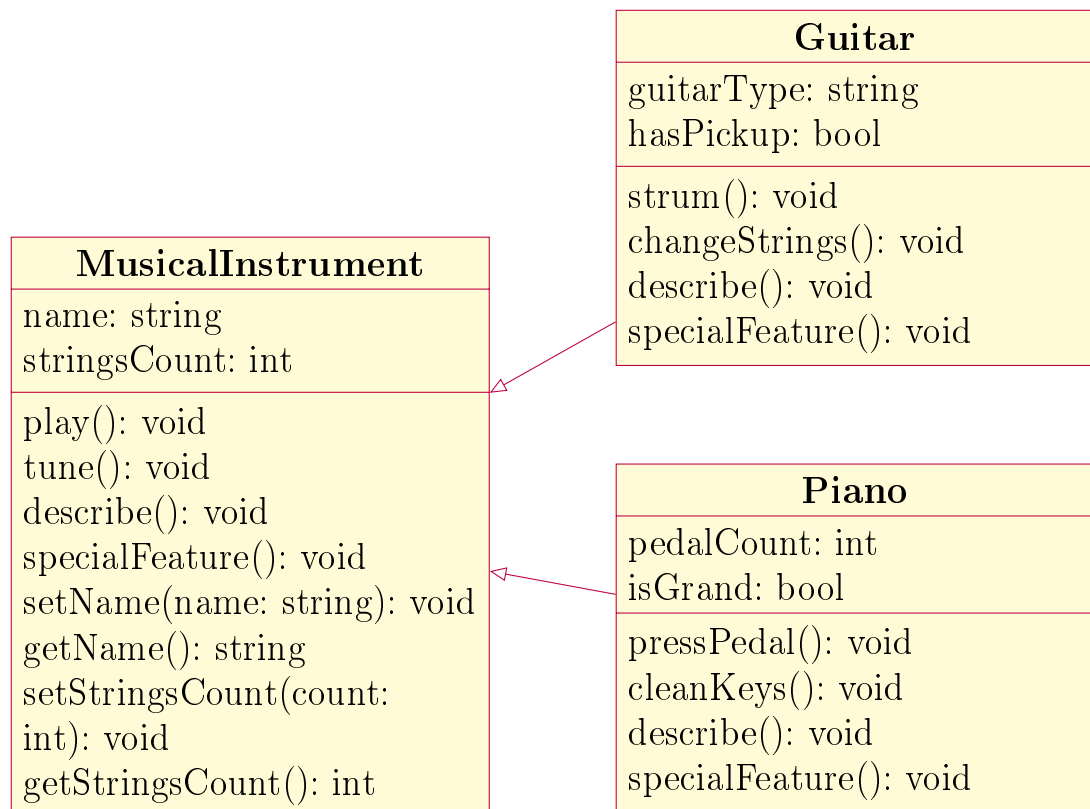


Рисунок 1 – Диаграмма классов

## 6 Вывод

В результате выполнения лабораторной работы были достигнуты следующие результаты:

- разработана иерархия классов с использованием принципов ООП;
- освоены техники наследования, переопределения методов и полиморфизма;
- реализована консольная система управления коллекцией музыкальных инструментов;
- создана UML-диаграмма, отражающая архитектуру программы.

Полученные знания и навыки будут полезны при дальнейшем изучении системного проектирования и архитектуры программного обеспечения.