



Рисунок 3 – Схема работы приложения

Литература

1. Шитиков Ю. А., Фесенко А. В. Создание мобильного приложения для Android OS в среде разработки Android Studio // Восемнадцатая всероссийская студенческая научно-практическая конференция Нижневартковского государственного университета. – 2016. – С. 1459–1463.
2. Пирская Л. Разработка мобильных приложений в среде Android Studio. – Litres, 2022.
3. Система управления зависимости Gradle [Электронный ресурс] URL: What is Gradle and why do we use it as Android developers? | by Van Markovic | Medium - Дата доступа 10.12.2023.

В. В. Никитенко

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **О. М. Дерюжкова**, канд. физ.-мат. наук, доцент

ШАБЛОН ПРОЕКТИРОВАНИЯ PROTOTYPE НА ЯЗЫКЕ C++

Прототип (Prototype) – это порождающий паттерн проектирования, который позволяет копировать объекты, не вдаваясь в подробности их реализации. Прототип поручает создание копий самим копируемым объектам. Для этого в базовый класс добавляется метод клонирования, который создает новый объект на основе существующего. Паттерн вводит общий интерфейс для всех объектов, поддерживающих клонирование. Это позволяет копировать объекты, не привязываясь к их конкретным классам.

Как это работает на C++?

В C++ реализация шаблона прототипа может быть сделана, используя методы клонирования или копирования.

Реализация шаблона начинается с определения класса Shape. Данный класс является прототипом для геометрических фигур в разрабатываемой программе. Он объявляет две основные виртуальные функции (рисунок 1).

```
C++
class Shape {
public:
    virtual Shape* clone() const = 0; // Clone method for creating copies.
    virtual void draw() const = 0; // Draw method for rendering the shape.
    virtual ~Shape() {} // Virtual destructor for proper cleanup.
};
```

Рисунок 1 – Класс Shape

Далее определяют классы Circle (рисунок 2) и Rectangle (рисунок 3). Данные классы наследуются от базовых классов и реализуют методы clone() и draw().

```
class Circle : public Shape {
private:
    double radius;
public:
    Circle(double r) : radius(r) {}

    Shape* clone() const override {
        return new Circle(*this);
    }

    void draw() const override {
        std::cout << "Drawing a circle with radius " << radius << std::endl;
    }
};
```

Рисунок 2 – Класс Circle

```
class Rectangle : public Shape {
private:
    double width;
    double height;
public:
    Rectangle(double w, double h) : width(w), height(h) {}

    Shape* clone() const override {
        return new Rectangle(*this);
    }

    void draw() const override {
        std::cout << "Drawing a rectangle with width " << width << " and height " << height << std::endl;
    }
};
```

Рисунок 3 – Класс Rectangle

Следующий этап – создание прототипов Circle и Rectangle (рисунок 4).

```
Circle circlePrototype(5.0);
Rectangle rectanglePrototype(4.0, 6.0);
```

Рисунок 4 – Создание прототипов

Далее эти прототипы необходимо клонировать для создания новых шаблонов (рисунок 5).

```
Shape* shape1 = circlePrototype.clone();  
Shape* shape2 = rectanglePrototype.clone();
```

Рисунок 5 – Клонирование

В конце используется метод draw для рендеринга фигур Circle и Rectangle (рисунок 6).

```
shape1->draw(); // Output: Drawing a circle with radius 5  
shape2->draw(); // Output: Drawing a rectangle with width 4 and height 6
```

Рисунок 6 – Вывод результата

Таким образом, шаблон Prototype с помощью копирования уже существующих прототипов позволяет эффективно создавать новые объекты. Он является полезным инструментом при разработке программ, благодаря своей гибкости и возможности повторно использовать код. Так же с его помощью можно значительно улучшить дизайн программного обеспечения. Шаблон Prototype позволяет создавать объекты с большей лёгкостью и не нарушая целостность кодовой базы.

Литература

1. Паттерны проектирования: какие бывают и как выбрать нужный [Электронный ресурс]. – 2023. – URL: <https://gb.ru/blog/patterny-proektirovaniya/>. – Дата доступа: 12.02.2024.
2. Прототип на C++ [Электронный ресурс]. – 2023. – URL: <https://refactoring.guru/ru/design-patterns/prototype/cpp/example/>. – Дата доступа: 19.02.2024.
3. Prototype Pattern | C++ Design Patterns [электронный ресурс]. – 2023. – URL: <https://www.geeksforgeeks.org/prototype-pattern-c-design-patterns/>. – Дата доступа: 19.02.2024.

Г. И. Озолинч

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. Н. Леванцов**, ст. преподаватель

РАЗРАБОТКА ОБРАБОТОК ПО ЗАГРУЗКЕ ДАННЫХ О КОНТАКТНОЙ И КОНФИДЕНЦИАЛЬНОЙ ИНФОРМАЦИИ КОНТРАГЕНТОВ «ІС:ПРЕДПРИЯТИЕ»

Обработки по загрузке какой-либо информации намного упрощают и ускоряют процесс введения информации в систему. Для передачи обработок третьим лицам используются внешние обработки.

При разработке обработок необходимо определить формат предоставленных данных. Информация предоставлялась в виде Excel-файлов разных видов размещения информации:

- на одном листе с разделением по заказчикам в строку;
- на разных листах с указанием названия листа как код заказчика в системе «ІС:Предприятие».