|  |
| --- |
| Изображение выглядит как текст, коллекция картинок, фарфор  Автоматически созданное описание  МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** |
| Институт кибербезопасности и цифровых технологий |
| Кафедра КБ-2 «Прикладные информационные технологии» |

**ЗАЧТЕНО**

\_\_\_\_\_\_\_\_\_\_ /Лесько С.А./

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_\_г.

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №5  
по дисциплине  
«Технологии и методы программирования»**

***(наименование дисциплины)***

**Выполнил**

Студент 3 курса:

Винницкий В.А.

Группы: БИСО-01-20

Специальности: 10.05.04

Москва 2023

**Оглавление**

[Абстрактная фабрика 3](#_Toc135411590)

[Строитель 3](#_Toc135411591)

[Адаптер 4](#_Toc135411592)

[Посредник 5](#_Toc135411593)

[Приложение 7](#_Toc135411594)

Абстрактная фабрика

На Рисунке 1 представлена UML-диаграмма реализации паттерна «Абстрактная фабрика», код диаграммы и исходный код реализации на C# представлены в GitHub-репозитории, который указан в Приложении.

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Рисунок 1 – Диаграмма для паттерна «Абстрактная фабрика»

*Описание.*

BankFactory - абстрактная фабрика, которая будет создавать конкретные объекты, IBankAccount, ICreditCard, ICreditProduct – интерфейсы сущностей, AlphaBankAccount, AlphaCreditCard, AlphaCreditProduct – конкретные сущности, BankClient - клиентский класс, который использует абстрактную фабрику для создания конкретных объектов

Строитель

На Рисунке 2 представлена UML-диаграмма реализации паттерна «Строитель», код диаграммы и исходный код реализации на C# представлены в GitHub-репозитории, который указан в Приложении.

Изображение выглядит как текст, снимок экрана, Шрифт, диаграмма

Автоматически созданное описание

Рисунок 2 – Диаграмма для паттерна «Строитель»

*Описание.*

BankAccount – класс банковского счета, IBankAccountBuilder - интерфейс строителя банковского аккаунта, BankAccountBuilder - класс строителя банковского аккаунта, который реализует интерфейс IBankAccountBuilder и поэтапно создает объекты BankAccount, BankAccountDirector - класс директора, который использует объекты строителя для создания BankAccount

Адаптер

На Рисунке 3 представлена UML-диаграмма реализации паттерна «Адаптер», код диаграммы и исходный код реализации на C# представлены в GitHub-репозитории, который указан в Приложении.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 3 – Диаграмма для паттерна «Адаптер»

*Описание.*

ThirdPartyBankAccount - "сторонний" класс, который предоставляет функциональность для работы с банковскими счетами, IBankAccount - интерфейс банковского счета, BankAccountAdapter - класс, который представляет адаптер между объектами ThirdPartyBankAccount и BankAccount

Посредник

На Рисунке 4 представлена UML-диаграмма реализации паттерна «Посредник», код диаграммы и исходный код реализации на C# представлены в GitHub-репозитории, который указан в Приложении.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 4 – Диаграмма для паттерна «Посредник»

*Описание.*

BankAccount - класс банковского счета, IBankMediator - интерфейс посредника, который определяет методы взаимодействия между объектами, BankMediator - класс посредника, реализует интерфейс.

Приложение

Ссылка на репозиторий https://github.com/Scythe888/TMP