|  |
| --- |
| Пермский филиал федерального государственного автономного образовательного учреждения высшего образования  «Национальный исследовательский университет  «Высшая школа экономики»  *Факультет социально-экономических и компьютерных наук* |
|  |
| Ханжин Александр Евгеньевич  **ЛАБОРАТОРНАЯ РАБОТА №13**  студента образовательной программы «Разработка информационных систем для бизнеса» по направлению подготовки *09.03.04 Программная инженерия*   |  |  | | --- | --- | |  | Старший преподаватель кафедры ИТБ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Марквирер В.Д. |   Пермь, 2024 |

# Оглавление

[Оглавление 2](#_Toc166604133)

[Задача 1 3](#_Toc166604134)

[1.1 Постановка задачи 3](#_Toc166604135)

[1.2 Диаграмма классов 3](#_Toc166604136)

[1.3 Алгоритм (блок-схема) 4](#_Toc166604137)

[1.4 Программа (листинг) 7](#_Toc166604138)

[1.5 Код тестов (листинг) 7](#_Toc166604139)

[1.6 Покрытие тестами кода 7](#_Toc166604140)

[Задача 2 8](#_Toc166604141)

[2.1 Постановка задачи 8](#_Toc166604142)

[2.2 Диаграмма классов 8](#_Toc166604143)

[2.3 Алгоритм (блок-схема) 9](#_Toc166604144)

[2.4 Программа (листинг) 11](#_Toc166604145)

[2.5 Код тестов (листинг) 11](#_Toc166604146)

[2.6 Покрытие тестами кода 11](#_Toc166604147)

[Задача 3 12](#_Toc166604148)

[3.1 Постановка задачи 12](#_Toc166604149)

[3.2 Диаграмма классов 12](#_Toc166604150)

[3.3 Алгоритм (блок-схема) 13](#_Toc166604151)

[3.4 Программа (листинг) 15](#_Toc166604152)

[3.5 Код тестов (листинг) 15](#_Toc166604153)

[3.6 Покрытие тестами кода 16](#_Toc166604154)

[Задача 4 17](#_Toc166604155)

[4.1 Постановка задачи 17](#_Toc166604156)

[4.2 Диаграмма классов 17](#_Toc166604157)

[4.3 Программа (листинг) 18](#_Toc166604158)

[4.4 Код тестов (листинг) 18](#_Toc166604159)

[4.5 Покрытие тестами кода 18](#_Toc166604160)

# Задача

## Постановка задачи

В программировании делегаты представляют собой мощный механизм, который позволяет реализовывать отслеживание изменений в объектах. В данной работе мы сосредоточимся на создании и использовании делегатов для отслеживания двух типов изменений в коллекциях: изменения количества элементов и изменения самих элементов.

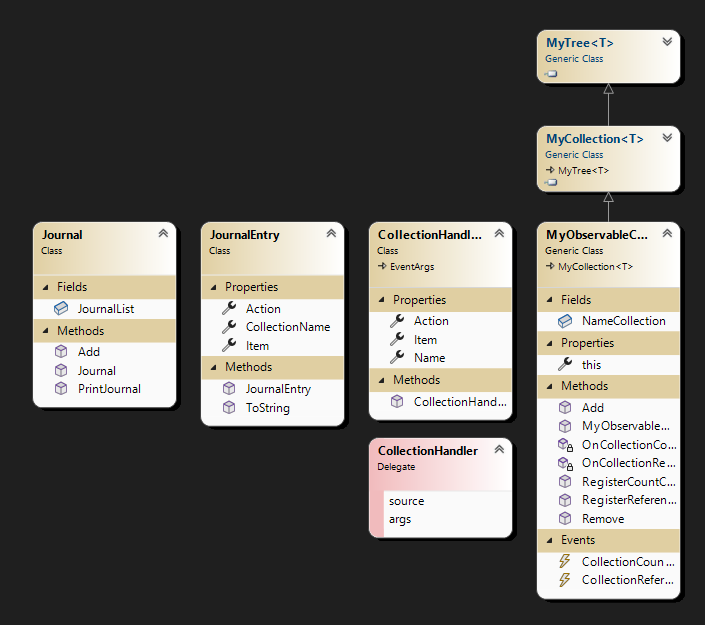
Делегаты в C# представляют собой типизированные ссылки на методы, что позволяет передавать методы как параметры, хранить и вызывать их в других методах. Мы собираемся создать делегат CollectionHandler, который будет использоваться для уведомления о любых изменениях в коллекции. Кроме того, мы определим класс CollectionHandlerEventArgs для передачи информации о типе изменений и объекте, вызвавшем изменение.

Наша цель - реализовать класс MyObservableCollection<T>, который будет наследоваться от MyCollection<T> и реализовывать методы для добавления, удаления, заполнения коллекции, а также бросать события при изменениях в коллекции. Мы создадим два события типа CollectionHandler - CollectionCountChanged для отслеживания изменений количества элементов и CollectionReferenceChanged для отслеживания изменений элементов в коллекции.

Для демонстрации работы делегатов и событий мы создадим две коллекции MyObservableCollection, два объекта Journal, которые будут подписаны на соответствующие события, и произведем изменения в коллекциях для последующего анализа данных в объектах Journal.

## Диаграмма классов

Диаграммы классов являются важным инструментом в объектно-ориентированном программировании для визуализации структуры классов и их взаимосвязей в программе или системе. Они представляют собой графическое представление классов, интерфейсов, атрибутов и методов, а также отношений между ними.



***Рис. 1.2.1. Диаграмма классов***

## Программа (листинг)

Полный код лабораторной работы можно посмотреть по ссылке https://github.com/Asklit/lab12/tree/master/lab13.

## Код тестов (листинг)

Полный код unit тестов списков можно посмотреть по ссылке https://github.com/Asklit/lab12/tree/master/lab13Test.

## Покрытие тестами кода

Для покрытия кода тестами я исключил все сторонние функции интерфейса, за счет чего получилось достичь 100% покрытия тестами кода.

