Лабораторная работа № 7

студента группы ИТз-221

Дмитриева Дмитрия Анатольевича

*Выполнение: \_\_\_\_\_\_\_\_\_ Защита: \_\_\_\_\_\_\_\_\_\_\_*

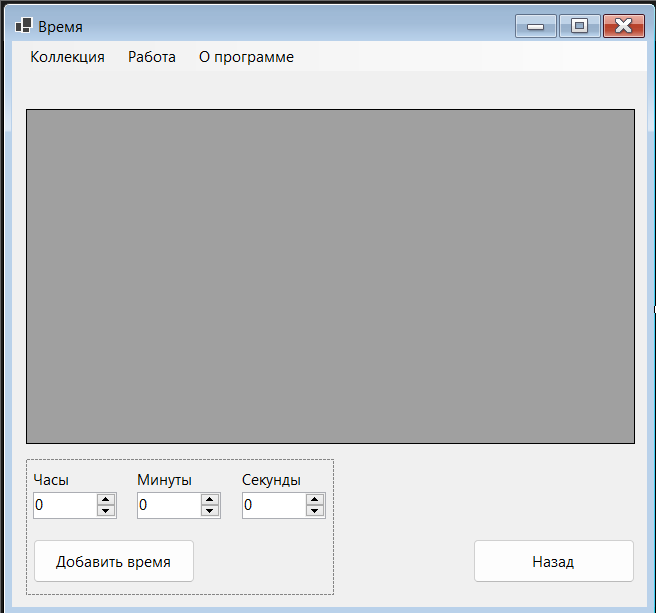
Массивы и коллекции C#

*Цель работы***:** знакомство с интерфейсами и классами коллекций C#. Получение навыков при работе с классом коллекции ArrayList. Освоение приемов организации работы в Windows Form приложении через меню.

**Ход работы:**

***Вариант - 12***

1. Разработал приложение с элементом управления MenuStrip. Используемые элементы на форме: Panel – для размещения элементов в определенном контейнере, DataGridView – для отображения массива времени, Button – обработчики, предназначенные для манипулирования массивом времени, NumericUpDown – для ввода значений времени от часов до секунд, Label – для подписи элементов ввода, MenuStrip – для отображения меню программы. Получил следующую форму (рис. 1)



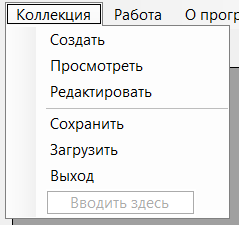
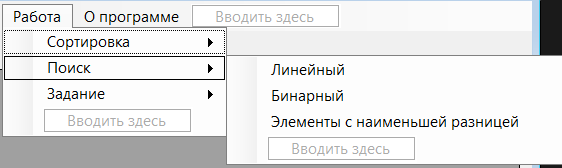
 

Рисунок 1 – главная форма приложения

1. Методы создания, редактирования и просмотра подразумевают под собой скрытие или отображение панели управления, а также создание нового массива времени в случае кнопки «Создать» (рис. 2).

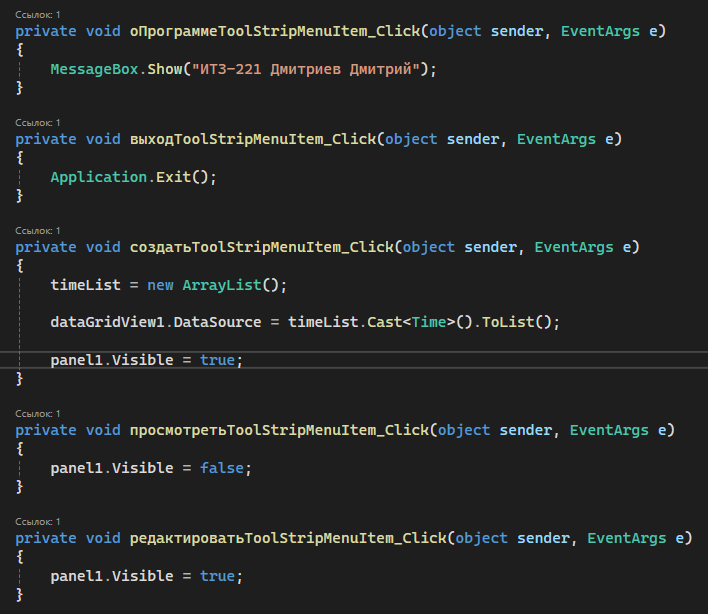
****

Рисунок 2 – методы MenuStrip

1. При нажатии на кнопку «Добавить время» считываются данные из полей ввода, а затем добавляется новый элемент в ArrayList (рис. 3).

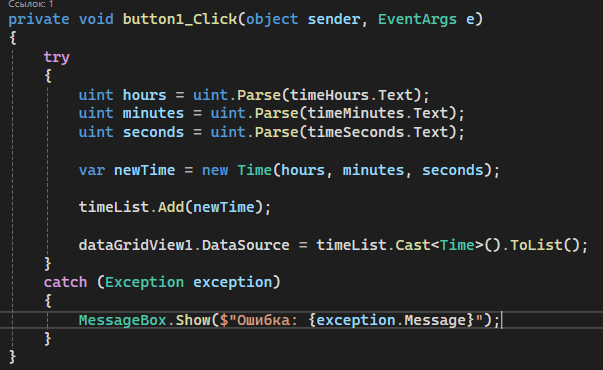


Рисунок 3 – метод добавления времени

1. Для сортировки и бинарного поиска создал отдельный класс TimeComparer, который наследует интерфейс IComparer, который позволяет использовать данный класс в виде обработчика сортировки ArrayList (рис. 4)

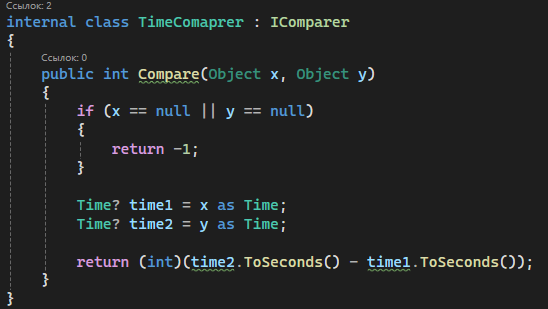


Рисунок 4 – класс для сортировки времени по убыванию

1. Добавил обработчики для линейного и бинарного поиска (рис. 5). В случае найденного элемента создается новый временный массив, куда помещаются найденные элементы.

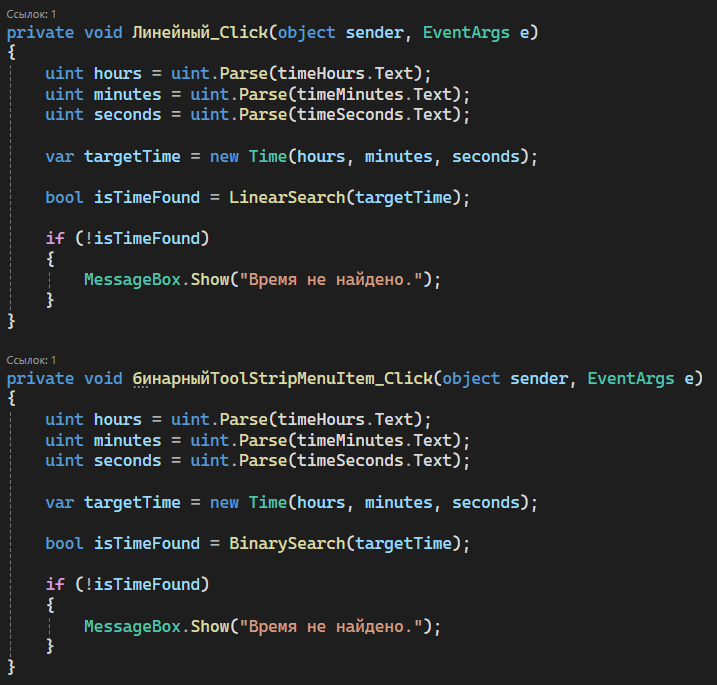


Рисунок 5 – обработчики поиска MenuStrip

1. Методы линейного (рис. 6) и бинарного поиска (рис. 7) соответственно.

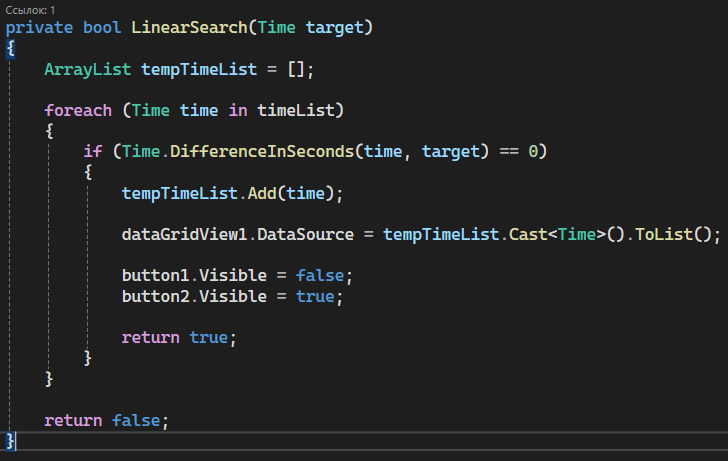


Рисунок 6 – линейный поиск

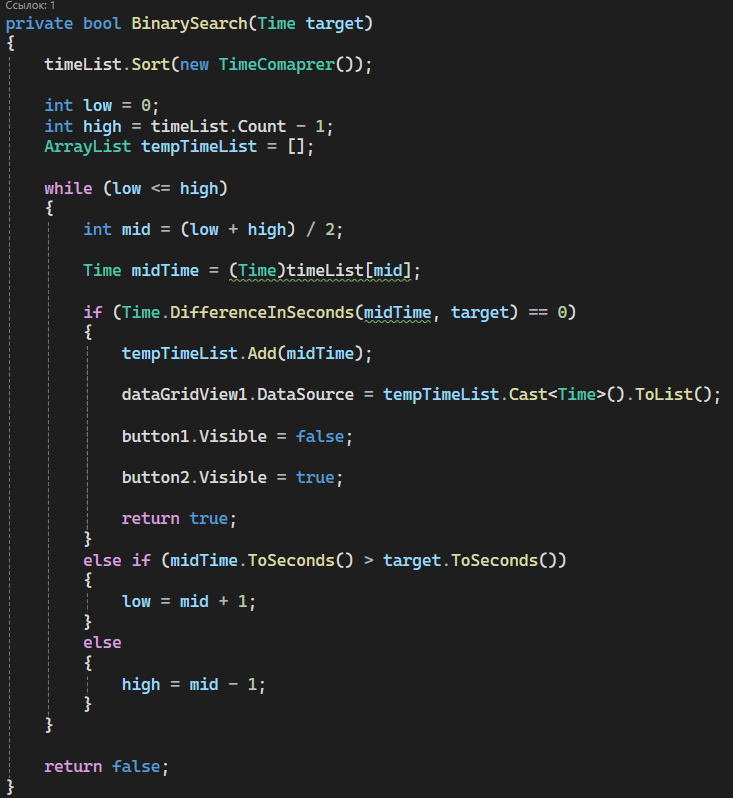


Рисунок 7 – бинарный поиск

1. Добавил метод обработчик для поиска двух элементов массива с наименьшей разницей по времени (рис. 8)

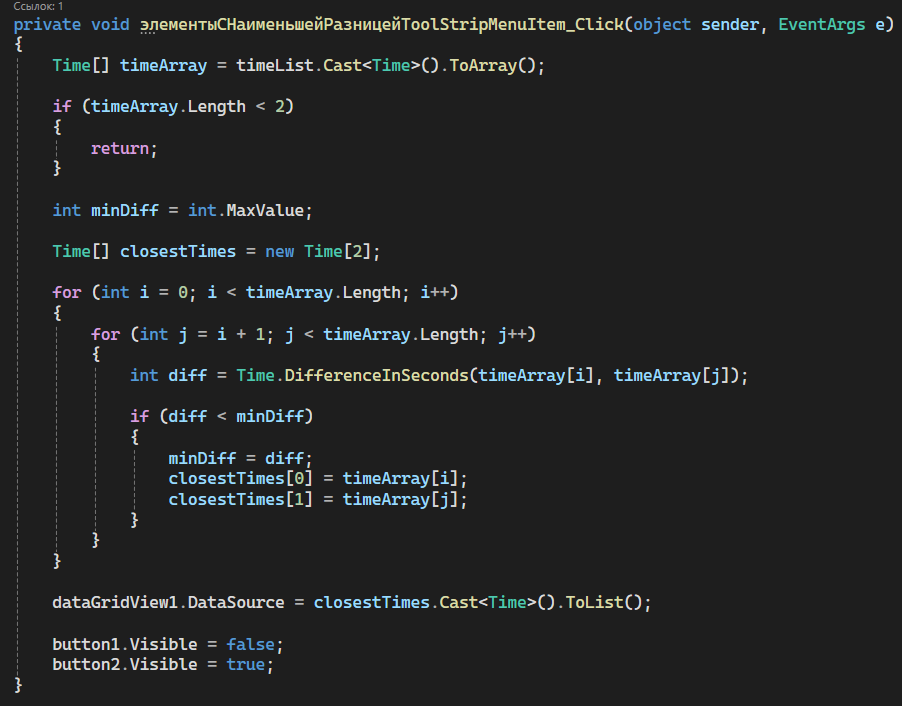
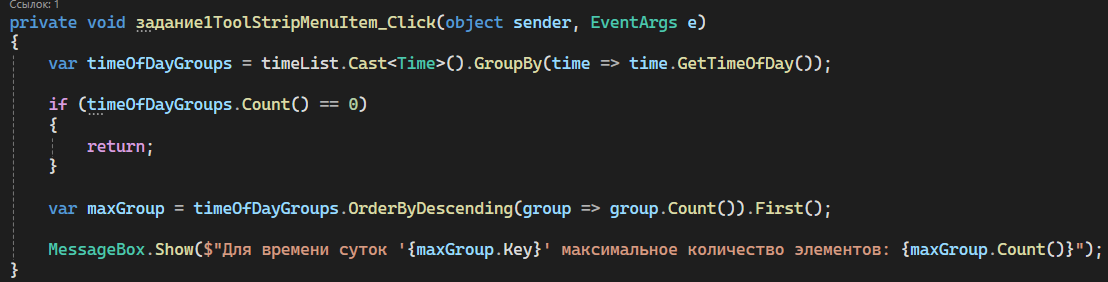


Рисунок 8 – поиск элементов с наименьшей разницей

1. Добавил метод обработчик индивидуального задания для определения для какого времени суток (утро, день, вечер, ночь) в коллекции максимальное количество элементов (рис. 9)



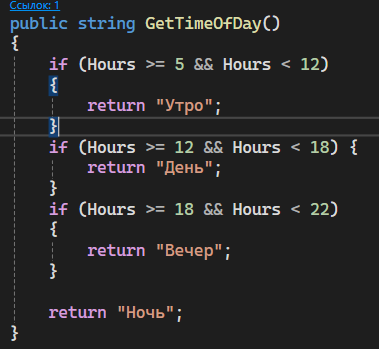
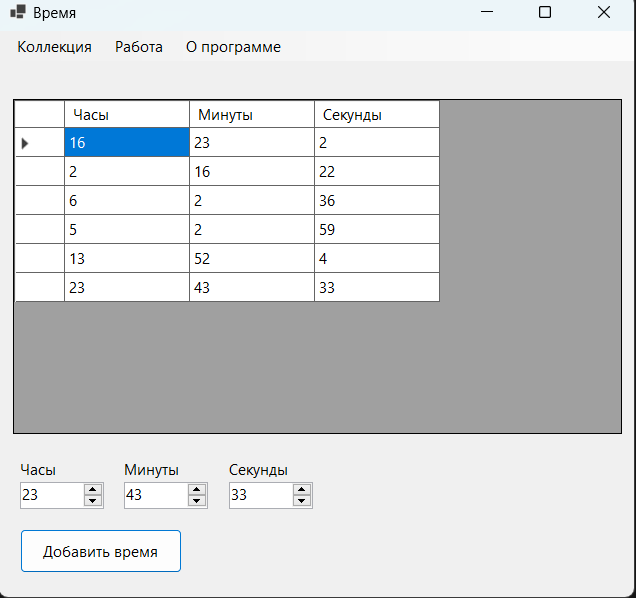
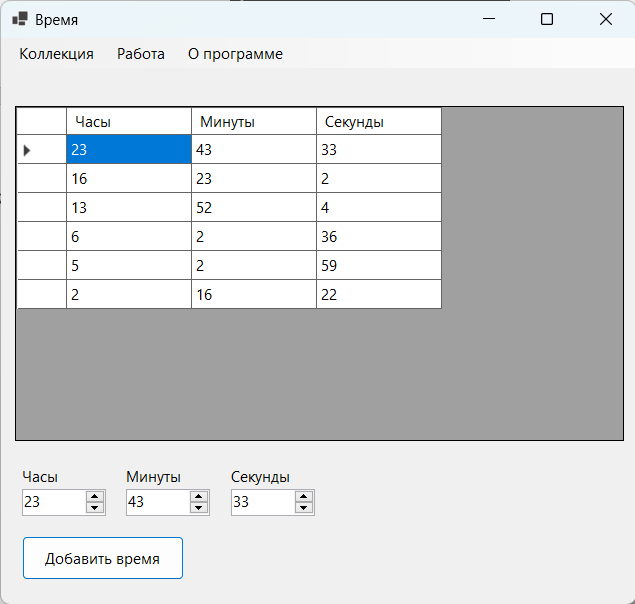


Рисунок 9 – индивидуальное задание

1. Протестировал программу, проверил индивидуальное задание, а также сортировку по убыванию (рис. 10).





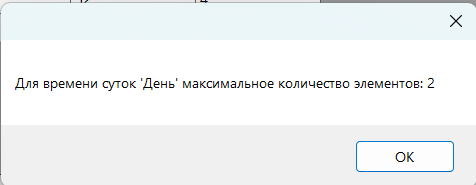


Рисунок 10 – результат выполнения программы

**Контрольные вопросы:**

1. Понятие и назначение интерфейса: Интерфейс в C# — это контракт, который определяет набор методов, которые класс должен реализовать. Он позволяет обеспечить полиморфизм и абстракцию.
2. Объявление интерфейса в C#: Интерфейс объявляется с использованием ключевого слова interface. Пример: public interface IDrawable { void Draw(); }.
3. Особенности реализации интерфейса в классах: Класс, реализующий интерфейс, должен предоставить реализацию всех его методов. Используется ключевое слово implements (или просто :, если интерфейс реализуется).
4. Понятие и назначение коллекций в C#: Коллекции в C# — это структуры данных, предназначенные для хранения и управления группами объектов. Они позволяют легко работать с динамическими наборами данных.
5. Типы коллекций в C#: В C# есть обобщенные и необобщенные коллекции, такие как List<T>, ArrayList, Dictionary<TKey, TValue>, Queue<T>, Stack<T> и другие.
6. Сравнение обобщенных и необобщенных коллекций: Обобщенные коллекции (List<T>, Dictionary<T, K>) обеспечивают типовую безопасность и лучшее выполнение. Необобщенные коллекции (ArrayList, Hashtable) могут содержать объекты различных типов, но менее эффективны.
7. Классы обобщенных коллекций: Классы обобщенных коллекций в C# включают List<T>, Dictionary<TKey, TValue>, Queue<T>, Stack<T>, и другие, предоставляющие типизированные структуры данных.
8. Интерфейсы, представляющие базовую функциональность коллекций в пространстве имен System.Collections: В System.Collections находятся интерфейсы ICollection, IEnumerable, IList, IDictionary, которые предоставляют базовую функциональность для работы с коллекциями.
9. Понятие и назначение коллекции ArrayList: ArrayList — это коллекция, которая может хранить элементы разных типов и динамически изменять размер. Она используется, когда неизвестно количество элементов заранее.
10. Интерфейсы, реализованные в классе System.Array: Класс Array реализует интерфейсы IEnumerable, ICloneable, IList, и другие, что позволяет работать с массивами как с коллекциями и предоставлять доступ к элементам через индексы.
11. Примеры методов класса System.Array: Методы класса Array включают Sort(), Reverse(), Copy(), Resize(), которые позволяют изменять порядок, копировать или изменять размер массивов.
12. Порядок работы с коллекцией ArrayList: Для работы с ArrayList нужно создать экземпляр класса, добавлять элементы методом Add(), получать доступ к элементам через индекс, и модифицировать коллекцию через методы, как Remove() или Clear().
13. Особенности сортировки и поиска элементов в коллекции ArrayList: В ArrayList можно сортировать элементы с помощью метода Sort() и искать элементы с помощью BinarySearch() или IndexOf().
14. Порядок создания главного меню в приложении: Для создания главного меню в Windows Forms используется элемент управления MenuStrip, в который добавляются элементы меню с помощью конструктора или кода.
15. Порядок создания контекстного меню в приложении: Контекстное меню создается через элемент ContextMenuStrip, а затем привязывается к нужному элементу управления с помощью свойства ContextMenuStrip.

**Вывод**: ознакомился с интерфейсами и классами коллекций C#. Получил навыки при работе с классом коллекции ArrayList. Освоил приемы организации работы в Windows Form приложения через меню.