Практическая работа №5

Обобщенные типы

Цель работы: изучить работу с обобщенными типами в языке C#.

Теоретическая часть

Обобщения (универсальные типы и методы, дженерики) позволяют разрабатывать классы и методы, которые откладывают спецификацию одного или нескольких типов до тех пор, пока класс или метод не будут объявлены и экземпляры не будут создаваться клиентским кодом.

Использование дженериков позволяет получить максимально широкие возможности многократного использования кода, обеспечения безопасности типов и повышения производительности.

Чаще всего универсальные шаблоны используются для создания классов коллекций. Библиотека классов .NET содержит несколько универсальных классов коллекций в пространстве имен `System.Collections.Generic`.

.NET позволяет создавать универсальные интерфейсы, классы, методы, события и делегаты.

Универсальные классы можно ограничить, чтобы они разрешали доступ к методам только для определенных типов данных.

Сведения о типах, используемых в универсальном типе данных, можно получить во время выполнения с помощью рефлексии.

Рассмотрим пример. Необходимо создать класс Tree для работы с бинарными деревьями. Добавим ограничение: тип данных в узле дерева должен реализовывать интерфейс IComparable.

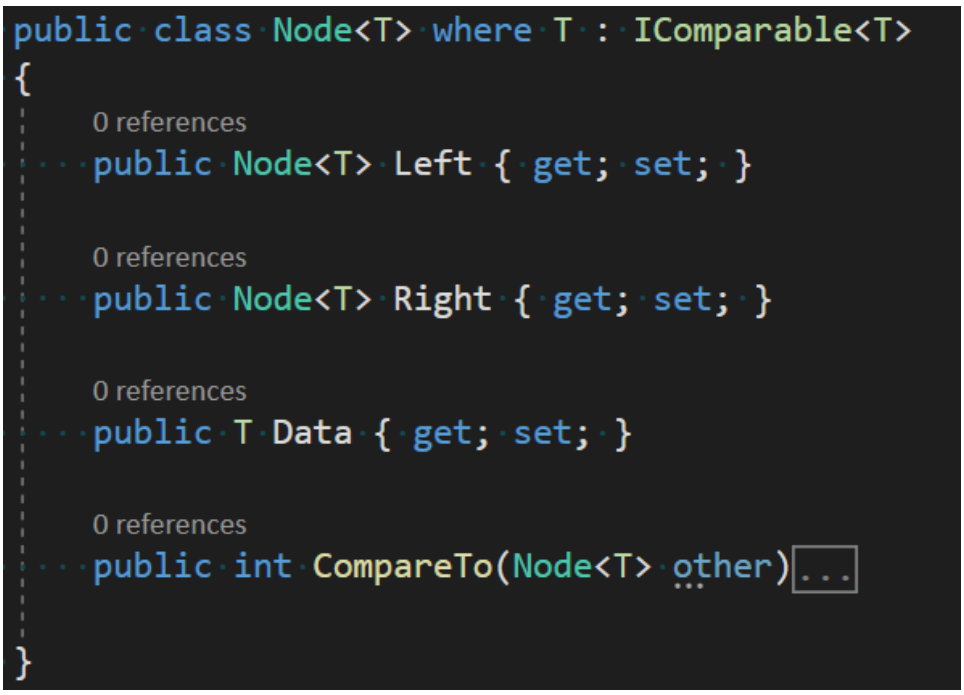

Рисунок 1 – Класс Node

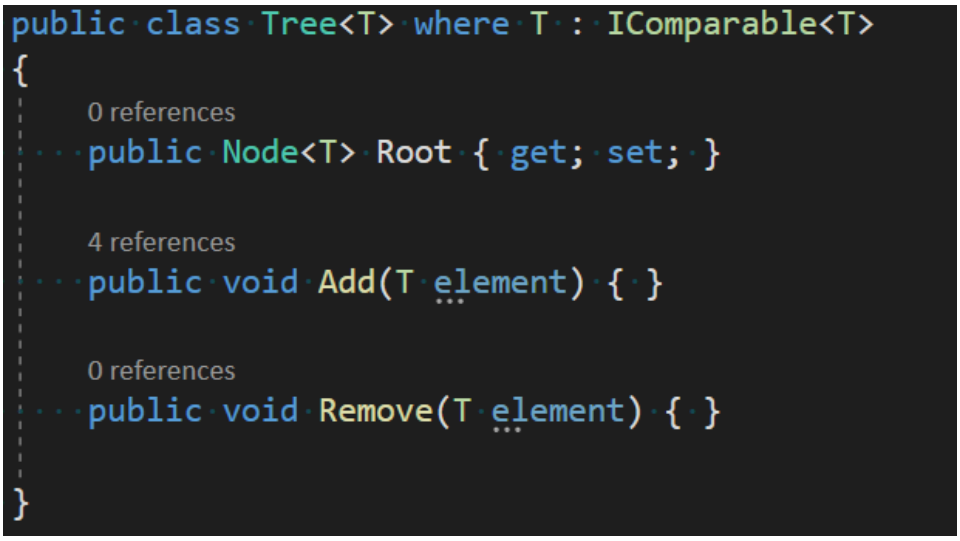

Рисунок 2 – Класс Tree

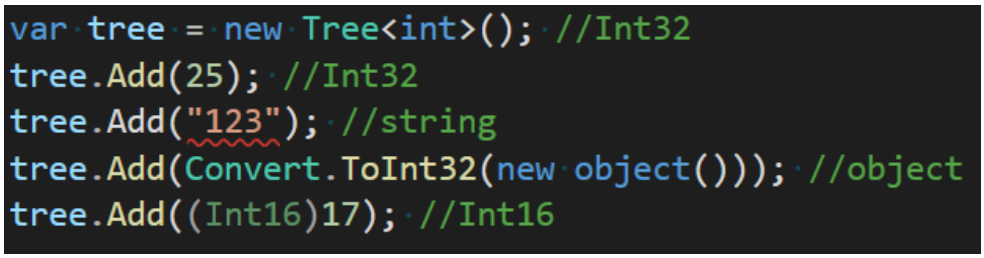

Рисунок 3 – Пример создания дерева

Практическая часть

1) Реализовать в классе `Figure` интерфейс `IComparable`. Сравнение фигур проводить по площади.
2) Создать класс `GenericList<T>`. Указать, что `T` должен реализовывать интерфейс `IComparable`. В классе создать приватное поле типа `List<T>`.
3) Добавить в класс `GenericList<T>` методы `Add(T)` и `BubbleSort()`. Метод должен сортировать элементы по возрастанию.
4) В консоли продемонстрировать работу с двумя экземплярами класса `GenericList<T>`. Элементами первого будут строки, а элементами второго – разнотипные фигуры (в одном списке квадраты, круги и т.д.).