# Проверяемое задание № 5

**Тема 5. Коллекции и обобщенные типы**

Лекция 5.2. Использование обобщенных методов и делегатов. Реализация пользовательского класса коллекции

**Цель работы**

Сформировать понимание и навыки использования базовых и обобщенных интерфейсов и классов коллекции .NET Framework. Научиться разрабатывать пользовательские классы коллекции, поддерживающие перечисление и инициализацию.

**Задание**

«**Бинарное дерево поиска**

1. Разработать тип «бинарное дерево поиска» для хранения результатов тестов, выполненных студентами, и их предоставления в упорядоченном виде.
2. Предусмотреть возможность хранения данных любого типа, поддерживающего сравнения.
3. Информация о студенте может содержать поля, хранящие имя студента, название теста, дату его прохождения и оценку теста для данного студента.

* Необходимо реализовать обобщенный стандартный интерфейс, чтобы информацию об одном студенте можно было сравнить с информацией о другом студенте.

1. Предусмотреть следующие конструкторы «дерева»:

* принимающий объект стандартного обобщённого типа для сравнения двух объектов одного типа и применяющий его для сравнения добавляемых в дерево элементов;
* принимающий объект типа стандартной обобщенной коллекции (не список!) и добавляющий элементы коллекции в дерево (пример смотрите в «исходниках» конструкторов типа List<T>);
* принимающий и коллекцию элементов для добавления в дерево, и объект, с помощью которого будут выполняться сравнения.

1. Итого: если указан «компаратор», то он используется для сравнения элементов, в противном случае используется «сравнение, реализованное внутри элемента».
2. Созданный класс «дерево» должен реализовывать обобщённый интерфейс перечислителя. При реализации нужно применить «итератор» (реализовать с помощью yield).
3. В «дереве» реализовать метод, который возвращает объект для итерации по дереву в обратном порядке.
4. Например, GetReversedEnumarator() (реализовать с помощью yield).
5. В качестве UI-интерфейса использовать консольное приложение с интерфейсом командной строки, WinForms или WPF-приложение. В приложении нужно:
   * Применить созданное «дерево» для хранения чисел из целочисленного массива, при этом в дереве они должны храниться в порядке убывания.
     + Добавить ещё несколько чисел в дерево.
     + Вывести все числа «дерева» в том порядке, в котором они храняться в дереве.
   * Применить созданное «дерево» для хранения результатов прохождения тестов студентами, при этом элементы в дереве должны храниться в порядке убывания «оценок (баллов)» по тесту.
     + Вывести информацию о студентах в том порядке, в котором они хранятся в дереве.
   * Применить созданное «дерево» для хранения строк:
     + Строки добавить из списка строк.
     + Вывести строки в обратном порядке, используя метод «дерева», который возвращает «перечислитель» для прохода «дерева» в обратном порядке». [1]

**Дополнительное задание (!)**

* Создать класс для итерации по дереву в обратном порядке (реализовать IEnumerator<T>).
  + Изменить в «дереве» метод, который возвращает объект для итерации по «дереву» в обратном порядке.

**Рекомендации по выполнению**

1. Само «дерево» не должно ничего выводить на «экран» (UI).
2. Если в дереве нельзя хранить элемент (значения) null, то нужно выбрасывать соответствующее исключение при попытке добавить элемент null.
3. yield (C# Reference): <http://msdn.microsoft.com/en-us/library/9k7k7cf0.aspx>
4. IEnumerator<T> Interface: <http://msdn.microsoft.com/en-us/library/78dfe2yb(v=vs.110).aspx>

**Задание 2**

**«Абстрактная фабрика»**

* Разработать иерархию классов для **произвольной предметной области** на основе представленной UML-диаграммы классов (рис. 1). Изучить отношения между классами и экземплярами классов.

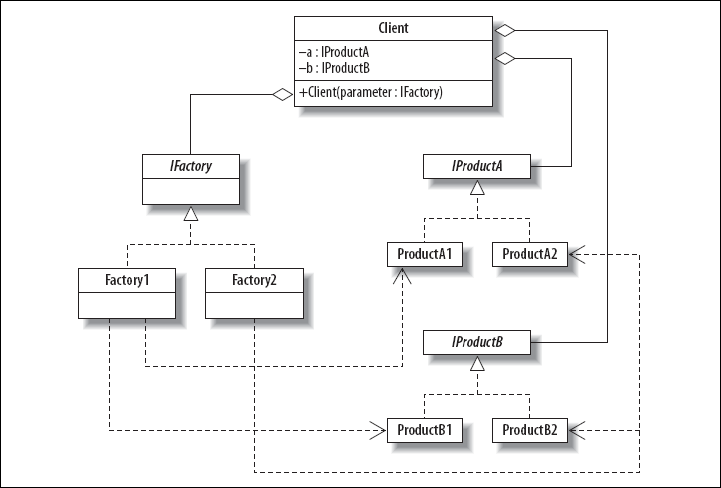


Рис. 1. UML-диаграмма

* У интерфейса IFactory должны быть методы CreateProductA(), возвращающий объект типа IProductA, и CreateProductB(), возвращающий объект типа IProductB.
* **Рекомендации по выполнению**

1. Названия классов и методов должны соответствовать **выбранной предметной области**. Примеры:
   * Мебель: стол, стул, диван, шкаф.
     + Плетёная мебель, офисная мебель, мебель для столовых.
   * Организация: рабочий, начальник, директор.
     + Завод, торговый ларёк, оптовый склад.
   * Работа с БД: соединение с БД, выполнение запроса в БД, параметр запроса БД.
     + MS SQL Server, Oracle, SQLite.

* **Важно (!)**

1. Обратите внимание на конструкторы и модификаторы доступа методов, конструкторов, свойств и полей, которые указаны на UML-диаграмме.

* В коде должно быть «один тип – один файл». Подробнее: <http://www.rsdn.ru/article/mag/200401/codestyle.xml#EIFAC>
* Для структурирования «классов» примените пространство имён. Подробнее: <http://www.rsdn.ru/article/mag/200401/codestyle.xml#E1DAC>

Framework Design Guidelines: Names of Namespaces –

<http://msdn.microsoft.com/en-us/library/vstudio/ms229026.aspx>

* + Должно быть общее пространство имён. В нём должны быть базовые интерфейсы. Пример: System.Data. **(!)**
  + Также должны быть «вложенные» пространства имён для конкретных объектов. Примеры: System.Data.SqlClient, System.Data.OracleClient.
* В качестве UI-интерфейса рекомендуется использовать консольное приложение с интерфейсом командной строки, WinForms или WPF-приложение.

**Требования к предоставлению результатов (!)**

**К отчету следует приложить:**

* скриншоты и краткое описание этапов работы;
* листинг программ, написанных на языке программирования C#.