

Лабораторная работа 3.

Универсальные типы. Классы-коллекции. Методы расширения класса `System.Linq.Enumerable`

Информация для всех вариантов

Во всех вариантах второго уровня требуется определить универсальный делегат

```
delegate System.Collections.Generic.KeyValuePair<TKey, TValue>  
GenerateElement<TKey, TValue>(int j);
```

и универсальный класс `TestCollections<TKey,TValue>`, который содержит закрытые поля следующих типов

- `System.Collections.Generic.List<TKey>`;
- `System.Collections.Generic.List<string>` ;
- `System.Collections.Generic.Dictionary<TKey, TValue>` ;
- `System.Collections.Generic.Dictionary<string, TValue>`;
- `GenerateElement<TKey, TValue>`.

Конкретные значения типовых параметров `TKey` и `TValue` зависят от варианта.

В конструкторе класса `TestCollections<TKey,TValue>` создаются коллекции с заданным числом элементов. Надо сравнить время поиска элемента в коллекциях-списках `List<TKey>` и время поиска элемента по ключу и элемента по значению в коллекциях-словарях `Dictionary<TKey,TValue>`.

Для автоматической генерации элементов коллекций надо определить метод, который принимает один целочисленный параметр типа `int` и возвращает ссылку на объект типа `KeyValuePair<TKey,TValue>`. Метод должен инициализировать объекты `KeyValuePair<TKey,TValue>` так, чтобы соответствие между номером элемента и объектом `TKey` в паре ключ-значение было взаимно-однозначным.

Метод для автоматической генерации элементов коллекций передается в класс `TestCollections<TKey,TValue>` через параметр конструктора класса. Для этого в классе `TestCollections<TKey,TValue>` надо определить конструктор с двумя параметрами, имеющими тип `int` и `GenerateElement`. Через целочисленный параметр объектам класса передается число элементов в коллекциях, через экземпляр делегата `GenerateElement` - метод, который используется для автоматической генерации пары ключ-значение в виде объекта `KeyValuePair<TKey,TValue>`.

Число элементов в коллекциях пользователь вводит в процессе работы приложения. Если при вводе была допущена ошибка, приложение должно

обработать исключение, сообщить об ошибке ввода и повторить прием ввода до тех пор, пока не будет правильно введено целочисленное значение.

Для четырех разных элементов - первого, центрального, последнего и элемента, не входящего в коллекцию, - надо измерить время поиска

- элемента в коллекциях `List<TKey>` и `List<string>` с помощью метода `Contains`;
- элемента по ключу в коллекциях `Dictionary< TKey, TValue>` и `Dictionary<string, TValue>` с помощью метода `ContainsKey`;
- значения элемента в коллекции `Dictionary< TKey, TValue>` с помощью метода `ContainsValue`.

Так как статический метод для автоматической генерации элементов должен обеспечивать взаимно-однозначное соответствие между значением целочисленного параметра метода и объектами `TKey`, его можно использовать как при создании коллекций, так и для генерации элемента для поиска.

Вариант 1. Требования к программе

Определить новые версии классов **Exam** и **Student** из лабораторной работы 2.

В класс **Exam** добавить реализацию интерфейсов

- `System.IComparable` для сравнения объектов типа `Exam` по названию предмета;
- `System.Collections.Generic.IComparer<Exam>` для сравнения объектов типа `Exam` по оценке.

Определить **вспомогательный класс**, реализующий интерфейс

`System.Collections.Generic.IComparer<Exam>`, который можно использовать для сравнения объектов типа `Exam` по дате экзамена.

В новой версии класса **Student** для списков зачетов и экзаменов использовать типы

- `System.Collections.Generic.List<Test>` для списка зачетов;
- `System.Collections.Generic.List<Exam>` для списка экзаменов.

В новой версии класса **Student** сохранить все остальные поля, свойства и методы из предыдущей версии класса, внести необходимые исправления в код свойств и методов из-за изменения типов полей для списков.

В классе **Student** определить методы для сортировки списка экзаменов

- по названию предмета;
- по оценке;

- по дате экзамена.

Определить универсальный делегат

```
delegate TKey KeySelector<TKey>(Student st);
```

Определить универсальный класс **StudentCollection<TKey>**, содержащий коллекцию объектов **Student**, в котором для хранения коллекции используется тип **System.Collections.Generic.Dictionary<TKey, Student>**. Типовой параметр **TKey** универсального класса **StudentCollection<TKey>** определяет тип ключа в коллекции **Dictionary<TKey, Student>**.

Метод, который используется для вычисления ключа при добавлении элемента **Student** в коллекцию класса **StudentCollection<TKey>**, отвечает делегату **KeySelector<TKey>** и передается **StudentCollection<TKey>** через параметр единственного конструктора класса.

Класс **StudentCollection<TKey>** содержит

- закрытое поле типа **System.Collections.Generic.Dictionary<TKey, Student>**;
- закрытое поле типа **KeySelector<TKey>** для хранения экземпляра делегата с методом, вычисляющим ключ для объекта **Student**;
- конструктор с одним параметром типа **KeySelector<TKey>** ;
- метод **void AddDefaults ()**, с помощью которого можно добавить некоторое число элементов типа **Student** для инициализации коллекции по умолчанию;
- метод **void AddStudents (params Student[])** для добавления элементов в коллекцию **Dictionary<TKey, Student>**;
- перегруженную версию виртуального метода **string ToString()** для формирования строки, содержащей информацию обо всех элементах коллекции **Dictionary<TKey, Student>**, в том числе значения всех полей класса **Student**, включая список зачетов и экзаменов;
- метод **string ToShortString()**, который формирует строку с информацией обо всех элементах коллекции **Dictionary<TKey, Student>**, состоящую из значений всех полей, среднего балла, числа зачетов и экзаменов для каждого элемента **Student**, но без списка зачетов и экзаменов.

В классе **StudentCollection<TKey>** определить свойства и методы, выполняющие операции со словарем **Dictionary<TKey, Student>** с использованием методов расширения класса **System.Linq.Enumerable** и статические методы-селекторы, которые необходимы для выполнения соответствующих операций с коллекцией:

- свойство типа **double** (только с методом **get**), возвращающее максимальное значение среднего балла для элементов **Dictionary<TKey, Student>**; если в коллекции нет элементов, свойство

возвращает некоторое значение по умолчанию; для поиска максимального значения среднего балла надо использовать метод `Max` класса `System.Linq.Enumerable`;

- метод

`IEnumerable<KeyValuePair<TKey, Student>> EducationForm(Education value)`, возвращающий подмножество элементов коллекции

`Dictionary<TKey, Student>` с заданной формой обучения; для формирования подмножества использовать метод `Where` класса `System.Linq.Enumerable`;

- свойство типа

`IEnumerable<IGrouping<Education, KeyValuePair<TKey, Student>>>` (только с методом `get`), выполняющее группировку элементов коллекции

`Dictionary<TKey, Student>` в зависимости от формы обучения студента с помощью метода `Group` класса `System.Linq.Enumerable`.

В методе **Main()**

1. Создать объект `Student` и вызвать методы, выполняющие сортировку списка экзаменов `List<Exam>` по разным критериям, после каждой сортировки вывести данные объекта. Выполнить сортировку
 - по названию предмета;
 - по оценке;
 - по дате экзамена.
2. Создать объект типа `StudentCollection<string>`. Добавить в коллекцию несколько разных элементов типа `Student` и вывести объект `StudentCollection<string>`.
3. Вызвать методы класса `StudentCollection<string>`, выполняющие операции с коллекцией-словарем `Dictionary<TKey, Student>`, и после каждой операции вывести результат операции. Выполнить
 - вычисление максимального значения среднего балла для элементов коллекции; вывести максимальное значение;
 - вызвать метод `EducationForm` для выбора студентов с заданной формой обучения, вывести результат фильтрации;
 - вызвать свойство класса, выполняющее группировку элементов коллекции по форме обучения; вывести все группы элементов.
4. Создать объект типа `TestCollection<Person, Student>`. Ввести число элементов в коллекциях и вызвать метод для поиска первого, центрального, последнего и элемента, не входящего в коллекции. Вывести значения времени поиска для всех четырех случаев.

Вариант 2. Требования к программе

Определить новые версии классов **Article**, **Edition** и **Magazine** из лабораторной работы 2.

В класс **Article** добавить реализации интерфейсов

- `System.IComparable` для сравнения объектов типа `Article` по названию статьи;
- `System.Collections.Generic.IComparer<Article>` для сравнения объектов типа `Article` по фамилии автора.

Определить **вспомогательный класс**, реализующий интерфейс

`System.Collections.Generic.IComparer<Article>`, который можно использовать для сравнения объектов типа `Article` по рейтингу статьи.

В новой версии класса **Magazine** использовать типы

- `System.Collections.Generic.List<Person>` для списка редакторов журнала;
- `System.Collections.Generic.List<Article>` для списка статей в журнале.

В новых версиях **Edition** и **Magazine** сохранить все остальные поля, свойства и методы из предыдущей версии класса, внести необходимые исправления в код свойств и методов из-за изменения типов полей для списка редакторов журнала и списка статей.

В классе **Magazine** определить методы для сортировки списка статей

- по названию статьи;
- по фамилии автора;
- по рейтингу статьи.

Определить универсальный делегат

```
delegate TKey KeySelector<TKey>(Magazine mg);
```

Определить универсальный класс **MagazineCollection<TKey>**, содержащий коллекцию объектов типа `Magazine`, в котором для хранения коллекции используется тип `System.Collections.Generic.Dictionary<TKey, Magazine>`.

Типовой параметр `TKey` универсального класса `MagazineCollection<TKey>` определяет тип ключа в коллекции `Dictionary<TKey, Magazine>`.

Метод, который используется для вычисления ключа при добавлении элемента `Magazine` в коллекцию класса `MagazineCollection<TKey>`, отвечает делегату `KeySelector<TKey>` и передается `MagazineCollection<TKey>` через параметр единственного конструктора класса.

Класс **MagazineCollection<TKey>** содержит

- закрытое поле типа `System.Collections.Generic.Dictionary<TKey, Magazine>`;
- закрытое поле типа `KeySelector<TKey>` для хранения экземпляра делегата с методом, вычисляющим ключ для объекта `Magazine`;

- конструктор с одним параметром типа `KeySelector<TKey>` ;
- метод `void AddDefaults()`, с помощью которого можно добавить некоторое число элементов типа `Magazine` для инициализации коллекции по умолчанию;
- метод `void AddMagazines (params Magazine[])` для добавления элементов в коллекцию `Dictionary<TKey, Magazine>`;
- перегруженную версию виртуального метода `string ToString()` для формирования строки, содержащей информацию обо всех элементах коллекции `Dictionary<TKey, Magazine>`, в том числе значения всех полей, включая список редакторов издания и список статей в журнале для каждого элемента `Magazine`;
- метод `string ToShortString()`, который формирует строку с информацией обо всех элементах коллекции `Dictionary<TKey, Magazine>`, содержащую значения всех полей, значение среднего рейтинга статей, число редакторов издания и число статей в журнале для каждого элемента `Magazine`, но без списков редакторов и статей.

В классе **`MagazineCollection<TKey>`** определить свойства и методы, выполняющие операции со словарем `Dictionary<TKey, Magazine>` с использованием методов расширения класса `System.Linq.Enumerable` и статические методы-селекторы, которые необходимы для выполнения соответствующих операций с коллекцией:

- свойство типа `double` (только с методом `get`), возвращающее максимальное значение среднего рейтинга статей для элементов коллекции; если в коллекции нет элементов, свойство возвращает некоторое значение по умолчанию; для поиска максимального значения среднего рейтинга статей надо использовать метод `Max` класса `System.Linq.Enumerable`;
- метод `IEnumerable<KeyValuePair<TKey, Magazine>> FrequencyGroup(Frequency value)`, возвращающий подмножество элементов коллекции `Dictionary<TKey, Magazine>` с заданной периодичностью выхода журнала; для формирования подмножества использовать метод `Where` класса `System.Linq.Enumerable`;
- свойство типа `IEnumerable<IGrouping<Frequency, KeyValuePair<TKey, Magazine >>>` (только с методом `get`), выполняющее группировку элементов коллекции `Dictionary<TKey, Magazine>` в зависимости от периодичности выхода журнала с помощью метода `Group` класса `System.Linq.Enumerable`.

В методе **`Main()`**

1. Создать объект `Magazine` и вызвать методы, выполняющие сортировку списка `List<Article>` статей в журнале по разным критериям, после каждой сортировки вывести данные объекта. Выполнить сортировку
 - по названию статьи;
 - по фамилии автора;
 - по рейтингу статьи.
2. Создать объект `MagazineCollection<string>`. Добавить в коллекцию несколько разных элементов типа `Magazine` и вывести объект `MagazineCollection<string>`.
3. Вызвать методы класса `MagazineCollection<string>`, выполняющие операции с коллекцией-словарем `Dictionary<TKey, Magazine>`, и после каждой операции вывести результат операции. Выполнить
 - вычисление максимального значения среднего рейтинга статей для элементов коллекции;
 - вызвать метод `FrequencyGroup` для выбора журналов с заданной периодичностью выхода;
 - вызвать свойство класса, выполняющее группировку элементов коллекции по периодичности выхода; вывести все группы элементов.
4. Создать объект типа `TestCollection<Edition, Magazine>`. Ввести число элементов в коллекции и вызвать метод для поиска первого, центрального, последнего и элемента, не входящего в коллекции. Вывести значения времени поиска для всех четырех случаев.

Вариант 3. Требования к программе

Определить новые версии классов **Paper**, **Team** и **ResearchTeam** из лабораторной работы 2.

В класс **Paper** добавить реализацию интерфейсов

- `System.IComparable` для сравнения объектов типа `Paper` по дате выхода публикации;
- `System.Collections.Generic.IComparer<Paper>` для сравнения объектов типа `Paper` по названию публикации.

Определить **вспомогательный класс**, реализующий интерфейс

`System.Collections.Generic.IComparer<Paper>`, который можно использовать для сравнения объектов типа `Paper` по фамилии автора публикации.

В новой версии класса **ResearchTeam** использовать типы

- `System.Collections.Generic.List<Person>` для списка участников проекта;

- `System.Collections.Generic.List<Paper>` для списка публикаций.

В новых версиях классов **Team** и **ResearchTeam** сохранить все остальные поля, свойства и методы из предыдущих версий, внести необходимые исправления в код свойств и методов из-за изменения типов полей для списка участников проекта и списка публикаций.

В классе **ResearchTeam** определить методы для сортировки списка публикаций

- по дате выхода публикации;
- по названию публикации;
- по фамилии автора.

Определить универсальный делегат

```
delegate TKey KeySelector<TKey>( ResearchTeam rt) ;
```

Определить универсальный класс **ResearchTeamCollection<TKey>**, содержащий коллекцию объектов типа `ResearchTeam`, в котором для хранения коллекции используется тип `System.Collections.Generic.Dictionary<TKey, ResearchTeam>`.

Типовой параметр `TKey` универсального класса `ResearchTeamCollection<TKey>` определяет тип ключа в коллекции `Dictionary<TKey, ResearchTeam>`.

Метод, который используется для вычисления ключа при добавлении элемента `ResearchTeam` в коллекцию класса `ResearchTeamCollection<TKey>`, отвечает делегату `KeySelector<TKey>` и передается `ResearchTeamCollection<TKey>` через параметр единственного конструктора класса.

Класс **ResearchTeamCollection<TKey>** содержит

- закрытое поле типа `System.Collections.Generic.Dictionary<TKey, ResearchTeam>`;
- закрытое поле типа `KeySelector<TKey>` для хранения экземпляра делегата с методом, вычисляющим ключ для объекта `ResearchTeam`;
- конструктор с одним параметром типа `KeySelector<TKey>` ;
- метод `void AddDefaults ()`, с помощью которого можно добавить некоторое число элементов `ResearchTeam` для инициализации коллекции по умолчанию;
- метод `void AddResearchTeams (params ResearchTeam [])` для добавления элементов в коллекцию `Dictionary<TKey, ResearchTeam>`;
- перегруженную версию виртуального метода `string ToString()` для формирования строки, содержащей информацию обо всех элементах

коллекции Dictionary<TKey, ResearchTeam>, в том числе значения всех полей, включая список участников проекта и список публикаций для каждого элемента ResearchTeam;

- метод string ToShortString(), который формирует строку с информацией обо всех элементах коллекции Dictionary<TKey, ResearchTeam>, содержащую значения всех полей, число участников проекта и число публикаций для каждого элемента ResearchTeam, но без списков участников и публикаций.

В классе **ResearchTeamCollection<TKey>** определить свойства и методы, выполняющие операции со словарем Dictionary<TKey, ResearchTeam> с использованием методов расширения класса System.Linq.Enumerable и статические методы-селекторы, которые необходимы для выполнения соответствующих операций с коллекцией:

- свойство типа DateTime (только с методом get), возвращающее дату последней по времени выхода публикации среди всех элементов коллекции; если в коллекции нет элементов, свойство возвращает значение по умолчанию для типа DateTime; для поиска максимального значения среднего рейтинга статей надо использовать метод Max класса System.Linq.Enumerable;
- метод IEnumerable<KeyValuePair<TKey, ResearchTeam>> TimeFrameGroup (TimeFrame value), возвращающий подмножество элементов коллекции Dictionary<TKey, ResearchTeam> со значением продолжительности исследований, которое передается как параметр; для формирования подмножества использовать метод Where класса System.Linq.Enumerable;
- свойство типа IEnumerable<IGrouping<TimeFrame, KeyValuePair<TKey, ResearchTeam>>> (только с методом get), выполняющее группировку элементов коллекции Dictionary<TKey, ResearchTeam> в зависимости от продолжительности исследований с помощью метода Group класса System.Linq.Enumerable.

В методе **Main()**

1. Создать объект ResearchTeam и вызвать методы, выполняющие сортировку списка публикаций List<Paper> по разным критериям, после каждой сортировки вывести данные объекта. Выполнить сортировку
 - по дате выхода публикации;
 - по названию публикации;
 - по фамилии автора.
2. Создать объект ResearchTeamCollection<string>. Добавить в коллекцию несколько разных элементов ResearchTeam и вывести объект ResearchTeamCollection<string>.

3. Вызвать методы класса `ResearchTeamCollection<string>`, выполняющие операции с коллекцией-словарем `Dictionary<TKey, ResearchTeam>`, после каждой операции вывести результат операции. Выполнить
- поиск даты последней по времени выхода публикации среди всех элементов коллекции;
 - вызвать метод `TimeFrameGroup` для выбора объектов `ResearchTeam` с заданным значением продолжительности исследований;
 - вызвать свойство класса, выполняющее группировку элементов коллекции по значению продолжительности исследований; вывести все группы элементов из списка.
4. Создать объект типа `TestCollection<Team, ResearchTeam>`. Ввести число элементов в коллекциях и вызвать метод для поиска первого, центрального, последнего и элемента, не входящего в коллекции. Вывести значения времени поиска для всех четырех случаев.