Лабораторная работа 3. Варианты первого уровня

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

Информация для всех вариантов

Во всех вариантах лабораторной работы 3 требуется определить класс **TestCollections**, который содержит 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>.

Конкретные значения типовых параметров TKey и TValue зависят от варианта. Во всех вариантах тип ключа TKey и тип значения TValue связаны отношением базовый-производный. Во всех вариантах в классе TValue определено свойство, которое возвращает ссылку на объект типа TKey с данными, совпадающими с данными подобъекта базового класса (это свойство должно возвращать ссылку на объект типа TKey, а не ссылку на вызывающий объект TValue).

В конструкторе класса **TestCollections** 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 в классе TestCollections надо определить статический метод, который принимает один целочисленный параметр типа int и возвращает ссылку на объект типа TValue.

Каждый объект TValue содержит подобъект базового класса TKey. Соответствие между значениями целочисленного параметра метода и подобъектами TKey класса TValue должно быть взаимно-однозначным - равным значениям параметра должны отвечать равные объекты TKey и наоборот. Равенство объектов типа TKey трактуется так же, как это было сделано в лабораторной работе 2 при определении операций равенства объектов.

Все четыре коллекции содержат одинаковое число элементов. Каждому элементу из коллекции List<TKey> должен отвечать элемент в коллекции Dictionary<TKey, TValue> с равным значением ключа. Список List<string> состоит из строк, которые получены в результате вызова метода ToString() для объектов TKey из списка List<TKey>. Каждому элементу списка

List<string> отвечает элемент в коллекции-словаре Dictionary<string, TValue> с равным значением ключа типа string.

Число элементов в коллекциях вводится пользователем 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, этот метод можно использовать как при создании коллекций с большим числом элементов, так и для генерации элемента для поиска.

Вариант 1. Требования к программе

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

В класс **Person** добавить реализацию интерфейсов

- System.IComparable для сравнения объектов типа Person по полю с фамилией;
- System.Collections.Generic.IComparer<Person> для сравнения объектов типа Person по дате рождения.

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 зачетов и экзаменов.

Определить **вспомогательный класс**, реализующий интерфейс System.Collections.Generic.IComparer<Student>, который можно использовать для сравнения объектов типа Student по среднему баллу.
Определить класс **StudentCollection**, который содержит

- закрытое поле типа System.Collections.Generic.List<Student>;
- 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 список List<Student>;
- перегруженную версию виртуального метода string ToString() для формирования строки с информацией обо всех элементах списка List<Student>, включающую значения всех полей, список зачетов и экзаменов для каждого элемента Student;
- метод string ToShortString(), который формирует строку с информацией обо всех элементах списка List<Student>, содержащую значения всех полей, средний балл, число зачетов и число экзаменов для каждого элемента Student, но без списков зачетов и экзаменов.

В классе **StudentCollection** определить методы, выполняющие сортировку списка List<Student>

- по фамилии студента с использованием интерфейса IComparable, реализованного в классе Person;
- по дате рождения студента с использованием интерфейса IComparer<Person>, реализованного в классе Person;
- по среднему баллу с использованием интерфейса IComparer<Student>, реализованного во вспомогательном классе.

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

- свойство типа double (только с методом get), возвращающее максимальное значение среднего балла для элементов списка List<Student>; если в коллекции нет элементов, свойство возвращает некоторое значение по умолчанию; для поиска максимального значения среднего балла использовать метод Max класса System.Linq.Enumerable;
- свойство типа IEnumerable<Student> (только с методом get), возвращающее подмножество элементов списка List<Student> с формой обучения Education.Specialist; для формирования подмножества использовать метод Where класса System.Linq.Enumerable;

- метод List<Student> AverageMarkGroup(double value), который возвращает список, в который входят элементы Student из списка List<Student> с заданным значением среднего балла; для формирования списка использовать методы Group и ToList класса System.Linq.Enumerable.

Определить класс **TestCollections**, в котором в качестве типа TKey используется класс Person, а в качестве типа TValue - класс Student. Класс содержит закрытые поля с коллекциями типов

- System.Collections.Generic.List<Person>;
- System.Collections.Generic.List<string>;
- System.Collections.Generic.Dictionary <Person, Student>;
- System.Collections.Generic.Dictionary <string, Student>.

В классе **TestCollections** определить

- статический метод с одним целочисленным параметром типа int, который возвращает ссылку на объект типа Student и используется для автоматической генерации элементов коллекций;
- конструктор с параметром типа int (число элементов в коллекциях) для автоматического создания коллекций с заданным числом элементов;
- метод, который вычисляет время поиска элемента в списках List<Person> и List<string>, время поиска элемента по ключу и время поиска элемента по значению в коллекциях-словарях Dictionary<Person, Student> и Dictionary<string, Student>.

В методе **Main()**

1. Создать объект типа StudentCollection. Добавить в коллекцию несколько различных элементов типа Student и вывести объект StudentCollection.
2. Для созданного объекта StudentCollection вызвать методы, выполняющие сортировку списка List<Student> по разным критериям, и после каждой сортировки вывести данные объекта. Выполнить сортировку
    - по фамилии студента;
    - по дате рождения;
    - по среднему баллу.
3. Вызвать методы класса StudentCollection, выполняющие операции со списком List<Student>, и после каждой операции вывести результат операции. Выполнить
    - вычисление максимального значения среднего балла для элементов списка;
    - фильтрацию списка для отбора студентов с формой обучения

Education.Specialist;
- группировку элементов списка по значению среднего балла; вывести все группы элементов.

4. Создать объект типа TestCollections. Вызвать метод для поиска в коллекциях первого, центрального, последнего и элемента, не входящего в коллекции. Вывести значения времени поиска для всех четырех случаев. Вывод должен содержать информацию о том, к какой коллекции и к какому элементу относится данное значение.

Вариант 2. Требования к программе

Определить новые версии классов **Edition** и **Magazine** из лабораторной работы 2.
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 классов **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 и списка статей.

В класс **Edition** добавить реализацию

- интерфейса System.IComparable для сравнения объектов Edition по полю с названием издания;
- интерфейса System.Collections.Generic.IComparer<Edition> для сравнения объектов Edition по дате выхода издания.

Определить **вспомогательный класс**, реализующий интерфейс System.Collections.Generic.IComparer<Edition>, который можно использовать для сравнения объектов типа Edition по тиражу издания.
Определить класс **MagazineCollection**, который содержит

- закрытое поле типа System.Collections.Generic.List<Magazine>;
- метод void AddDefaults (), с помощью которого в список List<Magazine> можно добавить некоторое число элементов типа Magazine для инициализации коллекции по умолчанию;
- метод void AddMagazines (params Magazine []) для добавления элементов в список List<Magazine>;
- перегруженную версию виртуального метода string ToString() для формирования строки с информацией обо всех элементах списка List<Magazine>, в том числе значения всех полей, список редакторов

журнала и список статей в журнале для каждого элемента Magazine;

- виртуальный метод string ToShortString(), который формирует строку с информацией обо всех элементах списка List<Magazine>, содержащую значения всех полей, средний рейтинг статей, число редакторов журнала и число статей в журнале для каждого элемента Magazine, но без списков редакторов и статей.

В классе **MagazineCollection** определить свойства и методы, выполняющие сортировку списка List<Magazine>

- по названию издания с использованием интерфейса IComparable, реализованного в классе Edition;
- по дате выхода издания с использованием интерфейса IComparer<Edition>, реализованного в классе Edition;
- по тиражу издания с использованием интерфейса IComparer<Edition>, реализованного во вспомогательном классе.

В классе **MagazineCollection** определить методы, выполняющие операции со списком List<Magazine> с использованием методов расширения класса System.Linq.Enumerable и статические методы-селекторы, которые необходимы для выполнения соответствующих операций с коллекциями:

- свойство типа double (только с методом get), возвращающее максимальное значение среднего рейтинга статей для элементов списка List<Magazine>; 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;
- свойство типа IEnumerable<Magazine> (только с методом get), возвращающее подмножество элементов списка List<Magazine> с периодичностью выхода журнала Frequency.Monthly; для формирования подмножества использовать метод Where класса System.Linq.Enumerable;
- метод List<Magazine> RatingGroup(double value), который возвращает список, содержащий элементы Magazine из List<Magazine> со средним рейтингом статей, который больше или равен value; для формирования списка использовать методы Group и ToList класса System.Linq.Enumerable.

Определить класс **TestCollections**, в котором в качестве типа TKey используется класс Edition, а в качестве типа TValue - класс Magazine. Класс содержит закрытые поля с коллекциями типов

- System.Collections.Generic.List<Edition>;
- System.Collections.Generic.List<string>;
- System.Collections.Generic.Dictionary <Edition, Magazine>;

- System.Collections.Generic.Dictionary <string, Magazine>.

В классе **TestCollection** определить

- статический метод с одним целочисленным параметром типа int, который возвращает ссылку на объект типа Magazine и используется для автоматической генерации элементов коллекций;
- конструктор с параметром типа int (число элементов в коллекциях) для автоматического создания коллекций с заданным числом элементов;
- метод, который вычисляет время поиска элемента в списках List<Edition> и List<string>, время поиска элемента по ключу и время поиска элемента по значению в коллекциях-словарях Dictionary< Edition, Magazine> и Dictionary<string, Magazine>.

В методе **Main()**

1. Создать объект типа MagazineCollection. Добавить в коллекцию несколько элементов типа Magazine с разными значениями полей и вывести объект MagazineCollection.
2. Для созданного объекта MagazineCollection вызвать методы, выполняющие сортировку списка List<Magazine> по разным критериям, и после каждой сортировки вывести данные объекта. Выполнить сортировку
    - по названию издания;
    - по дате выхода издания;
    - по тиражу издания.
3. Вызвать методы класса MagazineCollection, выполняющие операции со списком List<Magazine>, и после каждой операции вывести результат операции. Выполнить
    - вычисление максимального значения среднего рейтинга статей для элементов списка; вывести максимальное значение;
    - фильтрацию списка для отбора журналов с периодичностью выхода Frequency.Monthly, вывести результат фильтрации;
    - группировку элементов списка по значению среднего рейтинга статей; вывести все группы элементов.
4. Создать объект типа TestCollections. Вызвать метод для поиска в коллекциях первого, центрального, последнего и элемента, не входящего в коллекции. Вывести значения времени поиска для всех четырех случаев. Вывод должен содержать информацию о том, к какой коллекции и к какому элементу относится данное значение.

Вариант 3. Требования к программе

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

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 типа полей для списков.

В новую версию класса **Team** добавить реализацию интерфейса System.IComparable для сравнения объектов Team по полю с номером регистрации.

В новую версию класса **ResearchTeam** добавить реализацию интерфейса интерфейса System.Collections.Generic.IComparer<ResearchTeam> для сравнения объектов ResearchTeam по названию темы исследований.

Определить **вспомогательный класс**, реализующий интерфейс System.Collections.Generic.IComparer<ResearchTeam>, который можно использовать для сравнения объектов типа ResearchTeam по числу публикаций.

Определить класс **ResearchTeamCollection**, который содержит

- закрытое поле типа System.Collections.Generic.List<ResearchTeam>;
- метод void AddDefaults (), с помощью которого в список List<ResearchTeam> можно добавить некоторое число элементов типа ResearchTeam для инициализации коллекции по умолчанию;
- метод void AddResearchTeams(params ResearchTeam[]) для добавления элементов в список List<ResearchTeam>;
- перегруженную версию виртуального метода string ToString() для формирования строки с информацией обо всех элементах списка List<ResearchTeam>, которая содержит значения всех полей, список участников проекта и список публикаций для каждого элемента ResearchTeam;
- виртуальный метод string ToShortString(), который формирует строку с информацией обо всех элементах списка List<ResearchTeam>, включающую значения всех полей, число участников проекта и число публикаций для каждого элемента ResearchTeam, но без списков

участников и публикаций. В классе **ResearchTeamCollection** определить методы, выполняющие сортировку списка List<ResearchTeam>

- по номеру регистрации с использованием интерфейса IComparable, реализованного в классе Team;
- по названию темы исследований с использованием интерфейса IComparer<ResearchTeam>, реализованного в классе ResearchTeam;
- по числу публикаций с использованием интерфейса IComparer<ResearchTeam>, реализованного во вспомогательном классе.

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

- свойство типа int (только с методом get), возвращающее минимальное значение номера регистрации для элементов списка List<ResearchTeam>; если в коллекции нет элементов, свойство возвращает некоторое значение по умолчанию; для поиска минимального значения номера регистрации надо использовать метод Min класса System.Linq.Enumerable;
- свойство типа IEnumerable<ResearchTeam> (только с методом get), возвращающее подмножество элементов списка List<ResearchTeam> с продолжительностью исследований TimeFrame.TwoYears; для формирования подмножества использовать метод Where класса System.Linq.Enumerable;
- метод List<ResearchTeam> NGroup(int value), который возвращает список, в который входят элементы ResearchTeam из списка List<ResearchTeam> с заданным числом участников исследования; для формирования списка использовать методы Group и ToList класса System.Linq.Enumerable.

Определить класс **TestCollections**, в котором в качестве типа TKey используется класс Team, а в качестве типа TValue - класс ResearchTeam. Класс содержит закрытые поля с коллекциями типов

- System.Collections.Generic.List<Team>;
- System.Collections.Generic.List<string>;
- System.Collections.Generic.Dictionary <Team, ResearchTeam>;
- System.Collections.Generic.Dictionary <string, ResearchTeam>.

В классе **TestCollections** определить

- статический метод с одним целочисленным параметром типа int, который возвращает ссылку на объект типа ResearchTeam и используется для автоматической генерации элементов коллекций;

- конструктор с параметром типа int (число элементов в коллекциях) для автоматического создания коллекций с заданным числом элементов;
- метод, который вычисляет время поиска элемента в списках List<Team> и List<string>, время поиска элемента по ключу и время поиска значения элемента в коллекциях-словарях Dictionary<Team, ResearchTeam> и Dictionary <string, ResearchTeam>.

В методе **Main()**

1. Создать объект типа ResearchTeamCollection. Добавить в коллекцию несколько элементов типа ResearchTeam с разными значениями полей и вывести объект ResearchTeamCollection.
2. Для созданного объекта ResearchTeamCollection вызвать методы, выполняющие сортировку списка List<ResearchTeam> по разным критериям, и после каждой сортировки вывести данные объекта. Выполнить сортировку
   - по номеру регистрации;
   - по названию темы исследований;
   - по числу публикаций.
3. Вызвать методы класса ResearchTeamCollection, выполняющие операции со списком List<ResearchTeam>, и после каждой операции вывести результат операции. Выполнить
   - вычисление минимального значения номера регистрации для элементов списка; вывести минимальное значение;
   - фильтрацию проектов с продолжительностью исследований TimeFrame.TwoYears, вывести результат фильтрации;
   - группировку элементов списка по числу публикаций; вывести все группы элементов из списка.
4. Создать объект типа TestCollections. Вызвать метод для поиска в коллекциях первого, центрального, последнего и элемента, не входящего в коллекции. Вывести значения времени поиска для всех четырех случаев. Вывод должен содержать информацию о том, к какой коллекции и к какому элементу относится данное значение.

Лабораторная работа 3. Варианты второго уровня

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 Enumerable<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 Enumerable<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
   - поназванию статьи;
   - по фамилииавтора;
   - 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 коллекциях 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 значениию 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.