Федеральное государственное бюджетное образовательное учреждение высшего образования «Пермский национальный исследовательский политехнический университет»

Лабораторная работа №13

По дисциплине

“Объектно-ориентированное программирование”

Тема

“Разработка программы, управляемой событиями”

Вариант 13

Выполнил работу

студент группы РИС-19-1б

Мазитов Т.Э.

Проверила

доцент кафедры ИТАС

Викентьева О.Л.

Работу выполнил:

Пермь 2021

**Постановка задачи**

Создать иерархию классов (см. лаб. 10). Для каждого класса реализовать конструктор без параметров, с параметрами, свойства для доступа к полям объектов, метод для автоматического формирования объектов. Перегрузить метод ToString() для формирования строки со значениями всех полей класса.

Создать класс MyCollection как производный класс от класса Сollection<MyClass>.

Класс Сollection<MyClass> взять из лабораторной работы №12. В классе должны быть реализованы в классе методы для  заполнения коллекции (элементы коллекции формируются автоматически), добавления элементов коллекции, удаления элементов коллекции, сортировки элементов коллекции по заданному полю,  очистки коллекции, реализован итератор для доступа к элементам коллекции, реализовано свойство Length (только для чтения), содержащее текущее количество элементов коллекции.

Определить класс MyNewCollection производный от класса MyCollection, который с помощью событий извещает об изменениях в коллекции. Коллекция изменяется:

* при удалении/добавлении элементов
* при изменении одной из входящих в коллекцию ссылок, например, когда одной из ссылок присваивается новое значение.

В этом случае в соответствующих методах или свойствах класса бросаются события.

 В новую версию класса MyNewCollection добавить:

* открытое автореализуемое свойство типа string с названием коллекции;
* метод bool Remove (int j) для удаления элемента с номером j ; если в списке нет элемента с номером j, метод возвращает значение false;
* индексатор (с методами get и set) с целочисленным индексом для доступа к элементу с заданным номером.

 Для событий, извещающих об изменениях в коллекции, определяется свой делегат CollectionHandler с сигнатурой:

void CollectionHandler (object source, CollectionHandlerEventArgs args);

Для передачи информации о событии определить класс CollectionHandlerEventArgs, производный от класса System.EventArgs, который содержит

* открытое автореализуемое свойство типа string с названием коллекции, в которой произошло событие;
* открытое автореализуемое свойство типа string с информацией о типе изменений в коллекции;
* открытое автореализуемое свойство для ссылки на объект, с которым связаны изменения;
* конструкторы для инициализации класса;
* перегруженную версию метода string ToString() для формирования строки с информацией обо всех полях класса.

В класс MyNewCollection добавить два события типа CollectionHandler.

 CollectionCountChanged, которое происходит при добавлении нового элемента в коллекцию или при удалении элемента из коллекции; через объект CollectionHandlerEventArgs cобытие передает имя коллекции, строку с информацией о том, что в коллекцию был добавлен новый элемент или из нее был удален элемент, ссылку на добавленный или удаленный элемент;

 CollectionReferenceChanged, которое происходит, когда одной из ссылок, входящих в коллекцию, присваивается новое значение; через объект CollectionHandlerEventArgs событие передает имя коллекции, строку с информацией о том, что был заменен элемент в коллекции, и ссылку на новый элемент.

Событие CollectionCountChanged бросают следующие методы класса MyNewCollection

* AddDefaults();
* Add (object[] ) ;
* Remove (int index).

Событие CollectionReferenceChanged бросает метод set индексатора, определенного в классе MyNewCollection.

Информация об изменениях коллекции записывается в класс Journal, который хранит информацию в списке объектов типа JournalEntry. Каждый объект типа JournalEntry содержит информацию об отдельном изменении, которое произошло в коллекции. JournalEntry содержит:

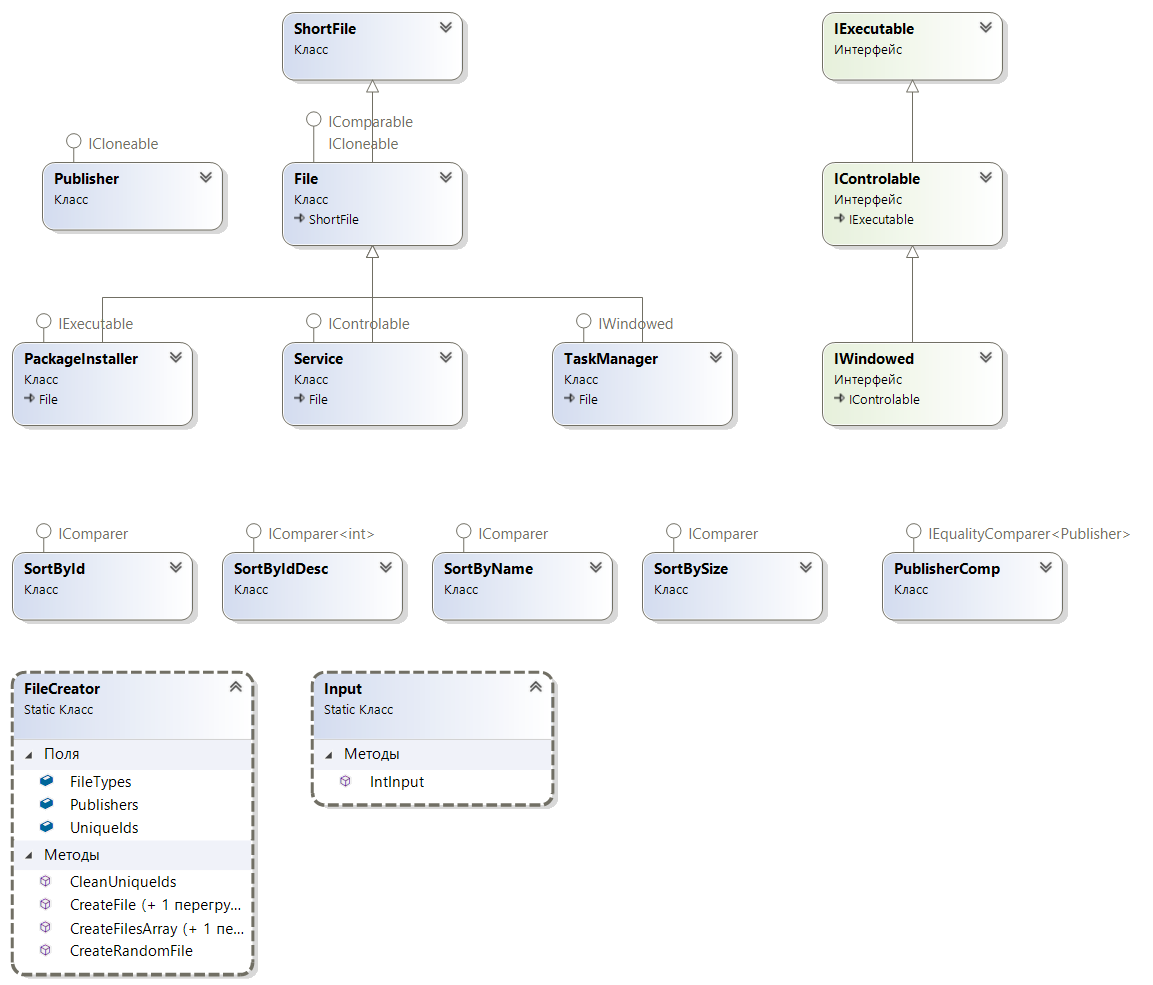
* открытое автореализуемое свойство типа string с названием коллекции, в которой произошло событие;
* открытое автореализуемое свойство типа string с информацией о типе изменений в коллекции;
* открытое автореализуемое свойство типа string c данными объекта, с которым связаны изменения в коллекции;
* конструктор для инициализации полей класса;
* перегруженную версию метода string ToString().
* всех элементах массива.

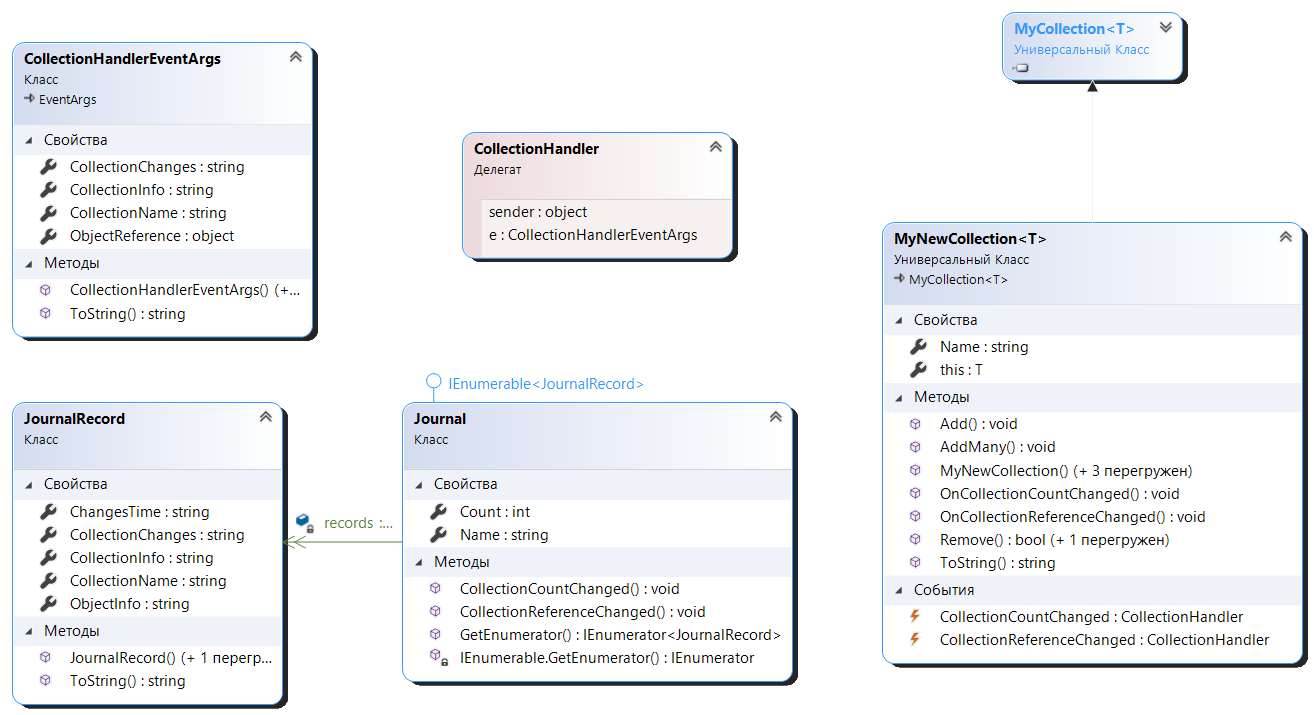
**Написать демонстрационную программу, в которой:**

* создать две коллекции MyNewCollection.
* Создать два объекта типа Journal, один объект Journal подписать на события CollectionCountChanged и CollectionReferenceChanged из первой коллекции, другой объект Journal подписать на события CollectionReferenceChanged из обеих коллекций.
* Внести изменения в коллекции MyNewCollection
* добавить элементы в коллекции;
* удалить некоторые элементы из коллекций;
* присвоить некоторым элементам коллекций новые значения.
* Вывести данные обоих объектов Journal.

**Содержание отчета:**

* Диаграмма классов.
* Определение делегата и событий.
* Определение функций, генерирующих события.
* Определение функций подписанных на событие.
* Операторы, которые выполняют подписку функции на событие.
* Текст демонстрационной программы.

**Проектирование**

****

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | | **Имя** | **Входные данные** | **Выходные данные** |
| **LR13class** | | | | |
| **Методы для работы с MyNewCollection<T>** | | | | |
|  | | Main () – входная точка программы | void | void |
|  | | Menu() | void | Вывод в консоль |
|  | | ShowCollections() | Коллекции | Вывод в консоль |
|  | | SetElementByIndex | void | Вывод в консоль |
|  | | GetElementByIndex | void | Вывод в консоль |
|  | | PrintCollection | Коллекция | Вывод в консоль |
|  | | RemoveAtIndex() | void | Вывод в консоль |
|  | | ChooseCollection() | void | collection - коллекция |
|  | | AddFile() | void | Вывод в консоль |
|  | | AddManyFiles() | void | Вывод в консоль |
|  | | CreateCollection() | void | Кортеж (имя коллекции, коллекция) |
| **Методы для работы с Journal** | | | | |
|  | CreateJournal() | | void | Кортеж (имя журнала, журнал) |
|  | ChooseJournal() | | void | journal - журнал |
|  | ShowJournals() | | Журналы | Вывод в консоль |
|  | PrintJournal() | | Журнал | Вывод в консоль |
|  | SubscribeJournal() | | void | Вывод в консоль |