Лабораторная работа № 1

по дисциплине

Объектно-ориентированное программирование

Общие требования к лабораторной работе

Определить класс Person, который имеет

- закрытое поле типа string, в котором хранится имя;
- закрытое поле типа string, в котором хранится фамилия;
- закрытое поле типа System.DateTime для даты рождения.

В классе Person определить конструкторы:

- конструктор с тремя параметрами типа string, string, DateTime для инициализации всех полей класса;
- конструктор без параметров, инициализирующий все поля класса некоторыми значениями по умолчанию.

В классе Person определить свойства с методами *get* и *set*:

- свойство типа string для доступа к полю с именем;
- свойство типа string для доступа к полю с фамилией;
- свойство типа DateTime для доступа к полю с датой рождения;
- свойство типа int с методами get и set для получения информации(get) и изменения (set) года рождения в закрытом поле типа DateTime, в котором хранится дата рождения.

В классе Person определить

- перегруженную(override) версию виртуального метода string ToString() для формирования строки со значениями всех полей класса;
- виртуальный метод string ToShortString(), который возвращает строку, содержащую только имя и фамилию.

В классе Person и в классах, дополнительно указанных в вариантах, надо

- переопределить (override) виртуальный метод bool Equals (object obj);
- определить операции == и != ;
- переопределить виртуальный метод int GetHashCode();

Реализация виртуального метода bool Equals (object obj) в классе System.Object определяет равенство объектов как равенство ссылок на объекты. Некоторые классы из базовой библиотеки BCL переопределяют метод Equals(). В классе System.String этот метод переопределен так, что равными считаются строки, которые совпадают посимвольно. Реализация метода Equals() в структурном типе DateTime определяет равенство объектов DateTime как равенство значений.

В лабораторной работе требуется переопределить метод Equals так, чтобы объекты считались равными, если равны все данные объектов. Для класса Person это означает, что равны даты рождения и посимвольно совпадают строки с именем и фамилией.

Определение операций == и != должно быть согласовано с переопределенным методом Equals, т.е. критерии, по которым проверяется равенство объектов в методе Equals, должны использоваться и при проверке равенства объектов в операциях == и !=.

Переопределение виртуального метода int GetHashCode() также должно быть согласовано с операциями == и !=. Виртуальный метод GetHashCode() используется некоторыми классами базовой библиотеки, например, коллекциями-словарями. Классы базовой библиотеки, вызывающие метод GetHashCode() из пользовательского типа, предполагают, что равным объектам отвечают равные значения хэш-кодов. Поэтому в

случае, когда под равенством объектов понимается совпадение данных (а не ссылок), реализация метода GetHashCode() должна для объектов с совпадающими данными возвращать равные значения хэш-кодов.

В классах, указанных в вариантах лабораторной работы, требуется определить метод object DeepCopy() для создания полной копии объекта.

Определенные в некоторых классах базовой библиотеки методы Clone() и Copy() создают ограниченную копию объекта – при копировании объекта копии создаются только для полей структурных типов, для полей ссылочных типов копируются только ссылки. В результате в ограниченной копии объекта поля-ссылки указывают на те же объекты, что и в исходном объекте.

Метод DeepCopy() должен создать полные копии всех объектов, ссылки на которые содержат поля типа. После создания полная копия не зависит от исходного объекта - изменение любого поля или свойства исходного объекта не должно риводить к изменению копии.

При реализации метода DeepCopy() в классе, который имеет поле типа System.Collections.ArrayList, следует иметь в виду, что определенные в классе ArrayList конструктор ArrayList(ICollection) и метод Clone() при создании копии коллекции, состоящей из элементов ссылочных типов, копируют только ссылки.

Метод DeepCopy() должен создать как копии элементов коллекции ArrayList, так и полные копии объектов, на которые ссылаются элементы коллекции. Для типов, содержащих коллекции, реализация метода DeepCopy() упрощается, если в типах элементов коллекций также определить метод DeepCopy().

Вариант 1.

Определить интерфейс

```
interface IDateAndCopy
{
        object DeepCopy();
        DateTime Date { get; set; }
}
```

В классе Person дополнительно

- переопределить метод virtial bool Equals (object obj) и определить операции == и != так, чтобы равенство объектов типа Person трактовалось как совпадение всех данных объектов, а не ссылок на объекты Person;
- переопределить виртуальный метод int GetHashCode();
- определить виртуальный метод object DeepCopy();
- реализовать интерфейс IDateAndCopy.

Определить тип Education - перечисление(enum) со значениями:

- Specialist
- Bachelor
- SecondEducation.

Определить класс Exam, который имеет три открытых автореализуемых свойства, доступных для чтения и записи:

- свойство типа string, в котором хранится название предмета;
- свойство типа int, в котором хранится оценка;
- свойство типа System.DateTime для даты экзамена.

В классе Exam определить:

- конструктор с параметрами типа string, int и DateTime для инициализации всех свойств класса;
- конструктор без параметров, инициализирующий все свойства класса некоторыми значениями по умолчанию;
- перегруженную(override) версию виртуального метода string ToString() для формирования строки со значениями всех свойств класса;
- реализовать интерфейс IDateAndCopy.

Определить класс Test, который имеет два открытых автореализуемых свойства, доступных для чтения и записи:

- свойство типа string, в котором хранится название предмета;
- свойство типа bool для информации о том, сдан зачет или нет.

В классе Test определить:

- конструктор с параметрами типа string и bool для инициализации свойств класса;
- конструктор без параметров, инициализирующий все свойства класса некоторыми значениями по умолчанию;
- перегруженную(override) версию виртуального метода string ToString() для формирования строки со значениями всех свойств класса;
- реализовать интерфейс IDateAndCopy.

Определить класс Student как производный от класса Person.
Класс Student имеет следующие поля:

- закрытое поле типа Education для информации о форме обучения;
- закрытое поле типа int для номера группы;
- закрытое поле типа System.Collections.ArrayList, в котором хранится список зачетов (объекты типа Test);
- закрытое поле типа Exam[] для информации об экзаменах, которые сдал студент.

В классе Student определить конструкторы:

- конструктор с параметрами типа Person, Education, int для инициализации соответствующих полей класса;
- конструктор без параметров для инициализации по умолчанию;

В классе Student определить свойства с методами get и set:

- свойство типа Person; метод get свойства возвращает объект типа Person, данные которого совпадают с данными подобъекта базового класса, метод set присваивает значения полям из подобъекта базового класса;
- свойство типа System.Collections.ArrayList с методами get и set для доступа к полю со списком зачетов;
- свойство типа Exam[] для доступа к полю со списком экзаменов.

В классе Student определить

- свойство типа double (только с методом get), в котором вычисляется средний балл как среднее значение оценок в списке сданных экзаменов;

- метод void AddExams (params Exam[]) для добавления элементов в список экзаменов;
- метод void AddTests (params Test[]) для добавления элементов в список тестов;
- перегруженная версия виртуального метода string ToString() для формирования строки со значениями всех полей класса, включая список зачетов и экзаменов;
- виртуальный метод string ToShortString(), который формирует строку со значениями всех полей класса без списка зачетов и экзаменов, но со значением среднего балла.

Дополнительно в классе Student

- определить перегруженную версию виртуального метода object DeepCopy();
- реализовать интерфейс IDateAndCopy;
- определить свойство типа int с методами get и set для доступа к полю с номером группы. В методе set бросить исключение, если присваиваемое значение меньше или равно 100 или больше 599. При создании объекта-исключения использовать один из определенных в библиотеке CLR классов-исключений, инициализировать объект-исключение с помощью конструктора с параметром типа string, в сообщении передать информацию о допустимых границах для значения свойства.

В отдельном методе Main()

1. Создать один объект типа Student, преобразовать данные в текстовый вид с помощью метода ToShortString() и вывести данные.
2. Присвоить значения всем определенным в типе Student свойствам, преобразовать данные в текстовый вид с помощью метода ToString() и вывести данные.
3. С помощью метода AddExams(params Exam[]) добавить элементы в список экзаменов и вывести данные объекта Student, используя метод ToString().
4. Создать два объекта типа Person с совпадающими данными и проверить, что ссылки на объекты не равны, а объекты равны, вывести значения хэш-кодов для объектов.
5. Создать объект типа Student, добавить элементы в список экзаменов и зачетов, вывести данные объекта Student.
6. Вывести значение свойства типа Person для объекта типа Student.
7. С помощью метода DeepCopy() создать полную копию объекта Student. Изменить данные в исходном объекте Student и вывести копию и исходный объект, полная копия исходного объекта должна остаться без изменений.
8. В блоке try/catch присвоить свойству с номером группы некорректное значение, в обработчике исключения вывести сообщение, переданное через объект-исключение.
9. С помощью оператора foreach вывести список всех зачетов и экзаменов.
10. С помощью оператора foreach для итератора с параметром вывести список всех экзаменов с оценкой выше 3.

Вариант 2.

Определить интерфейс

```
interface IRateAndCopy
{
        object DeepCopy();
        int Rating{ get; set; }
}
```

В классе Person дополнительно

- переопределить метод virtial bool Equals (object obj) и определить операции == и != так, чтобы равенство объектов типа Person трактовалось как совпадение всех данных объектов, а не ссылок на объекты Person;
- переопределить виртуальный метод int GetHashCode();
- определить виртуальный метод object DeepCopy();
- реализовать интерфейс IRateAndCopy.

Определить тип Frequency - перечисление(enum) со значениями:

- Weekly
- Monthly
- Yearly.

Определить класс Article, который имеет три открытых автореализуемых свойства, доступных для чтения и записи:

- свойство типа Person, в котором хранятся данные автора статьи;
- свойство типа string для названия статьи;
- свойство типа double для рейтинга статьи.

В классе Article определить:

- конструктор с параметрами типа Person, string, double для инициализации всех свойств класса;
- конструктор без параметров, инициализирующий все свойства класса некоторыми значениями по умолчанию;
- перегруженную(override) версию виртуального метода string ToString() для формирования строки со значениями всех свойств класса;
- определить виртуальный метод object DeepCopy();
- реализовать интерфейс IRateAndCopy.

Определить класс Edition. Класс Edition имеет

- защищенное(protected) поле типа string с названием издания;
- защищенное поле типа DateTime с датой выхода издания;
- защищенное поле типа int с тиражом издания;

В классе Edition определить:

- конструктор с параметрами типа string, DateTime, int для инициализации соответствующих полей класса;
- конструктор без параметров для инициализации по умолчанию;
- свойства с методами get и set для доступа к полям типа;
- виртуальный метод object DeepCopy();
- свойство типа int с методами get и set для доступа к полю с тиражом издания; в методе set свойства бросить исключение, если присваиваемое значение отрицательно. При создании объекта-исключения использовать один из определенных в библиотеке CLR классов-исключений, инициализировать объект-исключение с помощью конструктора с
- параметром типа string, в сообщении передать информацию о допустимых значениях свойства.

В классе Edition переопределить (override):

- виртуальный метод virtial bool Equals (object obj) и определить операции == и != так, чтобы равенство объектов типа Edition трактовалось как совпадение всех данных объектов, а не ссылок на объекты Edition;
- виртуальный метод int GetHashCode();
- перегруженную версию виртуального метода string ToString() для формирования строки со значениями всех полей класса.
- конструктор с параметрами типа string, Frequency, DateTime, int для инициализации соответствующих полей класса;
- конструктор без параметров, инициализирующий поля класса значениями по умолчанию.

Определить класс Magazine как производный от класса Edition.

Класс Magazine имеет базовый класс **Edition** и следующие поля:

- закрытое поле типа Frequency с информацией о периодичности выхода журнала;
- закрытое поле типа System.Collections.ArrayList со списком редакторов журнала (объектов типа Person);
- закрытое поле типа Article[] со списком статей в журнале.

В классе Magazine определить конструкторы:

- конструктор с параметрами типа string, Frequency, DateTime, int для инициализации соответствующих полей класса;
- конструктор без параметров для инициализации по умолчанию;

В классе Magazine определить свойства с методами *get* и *set*:

- свойство типа System.Collections.ArrayList для доступа к списку редакторов журнала;
- свойство типа Article[] для доступа к полю со списком статей;
- свойство типа double (только с методом get), в котором вычисляется среднее значение рейтинга статей в журнале;
- метод void AddArticles (params Article[]) для добавления элементов в список статей в журнале;
- метод void AddEditors (params Person[]) для добавления элементов в список редакторов;
- перегруженная версия виртуального метода string ToString() для формирования строки со значениями всех полей класса, включая список статей и список редакторов;
- виртуальный метод string ToShortString(), который формирует строку со значениями всех полей класса без списка статей и списка редакторов, но со значением среднего рейтинга статей в журнале.

Дополнительно в классе Magazine реализовать:

- перегруженную (override) версию виртуального метода object DeepCopy();
- интерфейс IRateAndCopy;
- свойство типа Edition; метод get свойства возвращает объект типа Edition, данные которого совпадают с данными подобъекта базового класса, метод set присваивает значения полям из подобъекта базового класса.

В отдельном методе Main()

- Создать один объект типа Magazine, преобразовать данные в текстовый вид с помощью метода ToShortString() и вывести данные.
- Присвоить значения всем определенным в типе Magazine свойствам, преобразовать данные в текстовый вид с помощью метода ToString() и вывести данные.
- С помощью метода AddArticles(params Article[]) добавить элементы в список статей и вывести данные объекта Magazine, используя метод ToString().
- Создать два объекта типа Edition с совпадающими данными и проверить, что ссылки на объекты не равны, а объекты равны, вывести значения хэш-кодов для объектов.
- В блоке try/catch присвоить свойству с тиражом издания некорректное значение, в обработчике исключения вывести сообщение, переданное через объект-исключение.
- Создать объект типа Magazine, добавить элементы в списки статей и редакторов журнала и вывести данные объекта Magazine.
- Вывести значение свойства типа Edition для объекта типа Magazine.
- С помощью метода DeepCopy() создать полную копию объекта Magazine. Изменить данные в исходном объекте Magazine и вывести копию и исходный объект, полная копия исходного объекта должна остаться без изменений.
- С помощью оператора foreach вывести список всех статей с рейтингом больше некоторого заданного значения.
- С помощью оператора foreach вывести список статей, в названии которых есть заданная строка.