ФГБОУ ВПО

«Дальневосточный государственный университет

путей сообщения»

Кафедра: «Информационные технологии и

системы»

Лабораторная работа №2

# «Наследование, исключения и итераторы»

Выполнил: Митряев В.Д.

Группа 23Б

Проверил: Латышевич Е.О.

Мельник А.К.

Хабаровск, 2015

**Задание:**

Определить интерфейс

interface IdateAndCopy

{

object DeepCopy();

DateTimeDate{ get; set; }

}

Определить новые версии классов **Exam**, **Person** и **Student** из задания 1. В эти классы добавить реализацию интерфейса IDateAndCopy. Новую версию класса Student определить как класс, производный от класса Person. Все поля новой версии класса Person определить с доступом protected, сохранить все свойства, определенные в первой версии класса.

В новой версии класса 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 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 c 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.

В классе Student определить:

* 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 System.Collections.ArrayList для списка экзаменов (*объекты типа Exam*).

Код следующих конструкторов, методов и свойств из старой версии класса Student необходимо изменить с учетом того, что часть полей класса перемещена в базовый класс Person, и в новой версии класса Student список экзаменов хранится в коллекции System.Collections.ArrayList:

* конструктор c параметрами типа Person, Education, int для инициализации соответствующих полей класса;
* конструктор без параметров для инициализациипо умолчанию;
* 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 double (только с методом get), в котором вычисляется средний балл как среднее значение оценок в списке сданных экзаменов;
* свойство типа System.Collections.ArrayList с методами get и set для доступа к полю со списком экзаменов;
* метод void AddExams(params Exam[]) для добавления элементов в список экзаменов;
* 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 новой версии класса 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 новой версии класса Student определить

* итератор для последовательного перебора всех элементов (объектов типа object) из списков зачетов и экзаменов (объединение);
* итератор c параметром для перебора экзаменов(объектов типа Exam) с оценкой больше заданного значения.

В методе Main()

1. 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.
2. Создать объект типа Student, добавить элементы в список экзаменов и зачетов, вывести данные объекта Student.
3. Вывести значение свойства типа Person для объекта типа Student.
4. 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.
5. В блоке try/catch присвоить свойству с номером группы некорректное значение, в обработчике исключения вывести сообщение, переданное через объект-исключение.
6. С помощью оператора foreach для итератора, определенного в классе Student, вывести список всех зачетов и экзаменов.
7. С помощью оператора foreach для итератора с параметром, определенного в классе Student, вывести список всех экзаменов с оценкой выше 3.

Ссылка на репозиторий:

https://github.com/Cl4pTrap/OOP/tree/mitryaev\_12/lab\_1

