

Лабораторна робота № 10 (6 балів)

Завдання 1

Створити масив об'єктів класу «Автобус» із елементами даних «марка», «кількість пасажирів», «максимальна швидкість», «ціна».

Перевантажити наступні операції: `==`, `!=`, `<`, `>`, `+`. Порівнювати в циклі попередній і поточний об'єкти і результати показувати на екрані.

Загальні вказівки

Не можна перевантажувати оператори:

`.` `=` `?` `sizeof` `&&` `||` `[]` `()` `new` `is`
`typeof` `+=` `-=` `*=` `/=`

- 1) Не можна змінювати пріоритети операцій.
- 2) Можна перевантажувати лише ті оператори, які існують в C#.
- 3) Оператори `<` та `>`, `==` та `!=`, `true` та `false` обов'язково перезавантажуються ПОПАРНО!
- 4) Унарний та бінарний варіанти операторів перевантажуються ОКРЕМО.

Синтаксис унарного оператора:

```
public static тип_результату operator ім'я(тип об'єкт)
{
}
```

Синтаксис бінарного оператора:

```
public static тип_результату operator ім'я(тип об'єкт1, тип об'єкт2)
{
}
```

Завдання 2

Визначити клас `Person`, який має

- закрите поле типу `string`, в якому зберігається ім'я;
- закрите поле типу `string`, в якому зберігається прізвище;
- закрите поле типу `System.DateTime` для дати народження.

В класі Person визначити конструктори:

- конструктор з трьома параметрами типу string, string, DateTime для ініціалізації всіх полів класу;
- конструктор без параметрів.

В класі Person визначити властивості з методами get та set:

- властивість типу string для доступу до поля з ім'ям;
- властивість типу string для доступу до поля з прізвищем;
- властивість типу DateTime для доступу до поля з датою народження;
- властивість типу int з методами get та set для отримання інформації (get) і зміни (set) року народження в закритому полі типу DateTime, в якому зберігається дата народження.

В класі Person визначити

- перевантажену (override) версію віртуального методу string ToString () для формування рядка зі значеннями всіх полів класу;
- віртуальний метод string ToShortString (), який повертає рядок, що містить тільки ім'я і прізвище.

Визначити тип Education - перерахування (enum) зі значеннями Specialist, Bachelor, SecondEducation.

Визначити клас Exam, який має три автоматичні властивості, доступні для читання і запису:

- властивість типу string, в якій зберігається назва предмета;
- властивість типу int, в якому зберігається оцінка;
- властивість типу System.DateTime для дати іспиту.

В класі Exam визначити:

- конструктор з параметрами типу `string`, `int` і `DateTime` для ініціалізації всіх властивостей класу;
- конструктор без параметрів;
- перевантажену (`override`) версію віртуального методу `string ToString ()` для формування рядка зі значеннями всіх властивостей класу.

Визначити клас `Student`, який має

- закрите поле типу `Person`, в якому зберігаються дані студента;
- закрите поле типу `Education` для інформації про форму навчання;
- закрите поле типу `int` для номера групи;
- закрите поле типу `Exam` для інформації про іспити, які здав студент.

В класі `Student` визначити конструктори:

- конструктор с параметрами типу `Person`, `Education`, `int` для ініціалізації відповідних полів класу;
- конструктор без параметрів.

В класі `Student` визначити властивості с методами `get` та `set`:

- властивість типу `Person` для доступу до поля з даними студента;
- властивість типу `Education` для доступу до поля з формою навчання;
- властивість типу `int` для доступу до поля з номером групи;
- властивість типу `Exam`, щоб отримати доступ до поля зі списком іспитів.

В класі `Student` визначити

- властивість типу `double` (тільки з методом `get`), в якому обчислюється середній бал як середнє значення оцінок в списку зданих іспитів;
- метод `void AddExams (params Exam [])` для додавання елементів в список іспитів;

- перевантажену версію віртуального методу `string ToString ()` для формування рядка зі значеннями всіх полів класу, включаючи список іспитів;
- віртуальний метод `string ToShortString ()`, який формує рядок зі значеннями всіх полів класу без списку іспитів, але зі значенням середнього балу.

В методі `Main ()`

- Створити один об'єкт типу `Student`, перетворити дані в текстовий вигляд за допомогою методу `ToShortString ()` і вивести дані.
- Вивести форму навчання (`Education.Specialist`, `Education.Bachelor` і `Education.SecondEducation`).
- Присвоїти значення всім визначеними в типі `Student` властивостями, перетворити дані в текстовий вигляд за допомогою методу `ToString ()` і вивести дані.
- С допомогою методу `AddExams (params Exam)` додати елементи в список іспитів і вивести дані об'єкта `Student`, використовуючи метод `ToString ()`.