Programowanie wizualne

opracował: Paweł Wojciechowski 2023/24

Lab 2

Zagadnienia do opanowania:

- typ `nullable`,
- klasy i obiekty
- właściwości,
- struktury readonly
- krotki

Temat: typ `nullable`

1. Zdefiniować zmienną `x` typu `Nullable<bool>`. Sprawdzić, czy poprawne jest wywołanie `if (x) {…}`.
2. Dla takiej klasy:

```
public class Boo
{
    public void ShowMessage() => Console.WriteLine("Hello!");
}
```

Przetestować działanie operatora „`??`” i „`?.`” Na poniższym przykładzie:

```
Boo boo = null;
boo.ShowMessage();
boo?.ShowMessage();
(boo ?? new Boo()).ShowMessage();
```

Pytanie kontrolne:

- Które wywołania ShowMessage() wyświetlą komunikat „Hello!”?
- czy obiekt dowolnej klasy może być typem `Nullable`?
- Jak wyłączyć ostrzeżenia CS8600? Na poziomie dyrektyw #nullable i ustawień projektu (<Nullable>).
- Jak uniknąć pojawienia się powyższego ostrzeżenia z wykorzystaniem operatora null forgiving (!).

Temat: Klasy i obiekty

2. Zdefiniować klasę abstrakcyjną Worker, która będzie miała następujące właściwości:
   • Imię
   • Nazwisko
   • ID (`System.Guid.NewGuid()`)
   • Rok urodzenia
3. Dodać metodę, `ToString()`, która wyświetli informacje o pracowniku.
4. Dodać metodę `GenerateNewID()`.
5. Zdefiniować konstruktor z parametrami.
6. Zdefiniować klasę `OfficeWorker`, która dziedziczy po `Worker` i jest klasą zapieczętowaną.
7. Stosując inicjalizację wartości właściwości utworzyć obiekt z wykorzystaniem inicjalizacji tak, jak na poniższym przykładzie:
   `OfficeWorker(x) { X = _x}`
8. Zdefiniować klasę `Manager`, która dziedziczy po `Worker` i przeciążyć metodę `ToString()`
   dodając informacje o roli „Manager”.
9. Zdefiniować klasę `Suprevisor`, która dziedziczy po `Manager` i przeciążyć metodę `ToString`() dodając informację o roli „Supervisor”. Przetestować kolejność wywoływania konstruktorów!
10. Zapieczętować metodę `ToString()` klasy `Manager` no i posprzątać (usunięcie ostrzeżenia CS0114).

11. Sprawdzić działanie statycznego konstruktora – zdefiniować i coś w nim wypisać. Utworzyć również konstruktor bezargumentowy – też coś w nim wypisać na konsoli. Czy w ciele konstruktora statycznego można utworzyć obiekt tej klasy?

    Pytania kontrolne:
    - jak zdefiniować właściwość automatyczną?
    - czy właściwość może być tylko do odczytu/zapisu?
    - czym się pola `readonly`?
    - ile konstruktorów statycznych można zdefiniować? Kiedy jest on uruchamiany?
    - co to jest i kiedy działa konstruktor domyślny?
    - jak wywołać jeden konstruktor z drugiego?

Temat: właściwości (Properies)

12. Zdefiniować klasę `Point` o dwóch właściwościach automatycznych X i Y typu int;
    - Sprawdzić, czy można używać set i init jednocześnie.
    - Czy właściwość w klasie może być `readonly` (tak jak poniżej)?
      ```
      public readonly int V { get; init; }
      ```
    - Zmienić `Point` z klasy na strukturę. Czy taka właściwość jak wyżej może zostać zdefiniowana?
    - Co się stanie, gdy do naszej struktury zostanie dodany konstruktor bezargumentowy?
    - Zmienić strukturę na `readonly` (całą strukturę). Czy klasa może być `readonly`?
    - Czy można zdefiniować następujące właściwości w ramach takiej struktury:
      - `public readonly int X1 { get; }`
      - `public readonly int X2 { get; set;}`
      - `public int Y { get; }`
      - `public int Z { get; init; }`
    - czym się różnią takie delkaracje:
      - `public readonly int X { get;}`
      - `public int Y { get; }`
      - `public int Z { get;  init; }`
    - dla tak przygotowanej struktury uruchomić ildasm (Menu: Narzędzia->Wiersz Polecenia->Program PowerShell dla Developerów). Otworzyć skompilowane assembly (bibliotekę dll) i sprawdzić na poziomie wygenerowanego kodu jakie są różnice.
    - Czy metoda w strukturze readonly może zmieniać wartości właściwości?

Temat: Krokti (tuples)

13. Przetestować sposoby tworzenia krotek i odwołania do poszczególnych elementów.
    ```
    (var Imie, var Nazwisko, var wiek) = ("Jan", "Nowak", 33);
    var (Imie2, Nazwisko2, wiek2) = ("Jan", "Nowak", 33);
    var pracownik = ("Jan", "Nowak", 33);
    (string Imie, string Nazwisko, int Wiek) prac = ("Jan", "Nowak",
    33);
    ```