

Programowanie komputerów.

Ćwiczenia 10.

INTERFEJSY, KOLEKCJE

Przed przystąpieniem do zadań należy przeanalizować treść wykładu na temat interfejsów oraz z rozdziałem 5. podręcznika „*Programowanie komputerów w zadaniach. Język C#*” (zadania 5.1-5.3).

Wstępnie przygotowany kod dla zadań 5.2-5.3 znajduje się w pliku `zad_0.txt`

Zadanie 1.

a)

Napisz program z użyciem dwóch klas i jednego interfejsu. Interfejs **IFigura** ma posiadać właściwość dla koloru figury (z użyciem typu `ConsoleColor`) i metodę `Pole`. Klasa **Kolo** ma zapamiętywać promień koła. Natomiast klasa **Prostokat** ma zapamiętywać długości boków. W obu klasach wykonaj implementacje interfejsu **IFigura**. Wyposaż obie klasy w metodę zwracającą tekst opisujący dane `ToString`. W metodzie **Main** zdefiniuj listę figur (jedną wspólną dla kół i prostokątów) i zaprezentuj dane o figurach.

b)

Do programu dodaj możliwość sortowania listy figur według powierzchni. W tym celu dokonaj rozbudowania interfejsu **IFigura** o interfejs **IComparable<T>** i odpowiednio zaimplementuj metodę `CompareTo` w obu klasach **Kolo** i **Prostokat**. W metodzie **Main** zdefiniuj listę figur (jedną wspólną dla kół i prostokątów) i wyświetl posortowaną listę figur według ich powierzchni.

Zadanie 2.

a)

Napisz program z klasą generyczną dla typów implementujących interfejs **IOsoba** (wykorzystaj w tym celu stosowne ograniczenie). Podobne zadanie było w poprzedniej prezentacji, ale bez udziału interfejsów. Interfejs **IOsoba** (z właściwościami `Imie` i `Nazwisko`) ma być rozbudowany o interfejs **IAdres** (z właściwościami opisującymi pozostałe dane o adresie). Program ma służyć do formatowania tekstu z adresami osób, jaki docelowo można by drukować na naklejkach. Stwórz w programie dwie klasy implementujące interfejs **IOsoba**, np. **Klient** i **Pracownik**, a następnie przetestuj program wyświetlając sformatowane adresy na ekranie dla kilku różnych obiektów umieszczonych w liście

b)

Do programu dodaj możliwość sortowania wykazu adresów według kryterium zadanego przez użytkownika, dając mu do wyboru takie dane jak nazwisko, imię, miasto, ulica.

Zadanie 3.

Wykonaj program stanowiący modyfikację przykładu dołączonego do prezentacji z wykładu (w pliku `przyklady_interfejs.txt`). W nowej wersji programu należy wykonać następujące zmiany:

- Zamiast składnika statycznego `podstawaPremii` – ma być właściwość publiczna wymagana przez interfejs **IPremia**. Należy odpowiednio zmienić konstruktory klas **Pracownik** i **Dyrektor**.
- Program ma umożliwić sortowanie pracowników według nazwiska i imienia. Wykorzystaj w tym celu interfejs **IComparable<T>**.
- Należy do programu dodać klasę **DyrektorNaczelny**. W klasie tej premia ma być liczona według algorytmu: premia dyrektorska (jak dla dyrektorów) + kwota specjalna, która ma być w tej klasie dodatkowym polem.

Zadanie 4.

Popraw ten program, tak aby działał. W programie tym stworzono własną kolekcję przy użyciu klasy **Kolory**. W metodzie **Main** ma być możliwe wykorzystanie pętli `foreach` dla tej kolekcji.

```
public class Kolory
{
    string[] teksty = null;
    public Kolory(params string[] tabParam)
    {
        teksty = new string[tabParam.Length];
        int indeks = 0;
        foreach (string txt in tabParam)
        {
            teksty[indeks++] = txt;
        }
    }
}
class Glowna
{
    static void Main(string[] args)
    {
        Kolory k = new Kolory("biały", "czerwony", "żółty", "zielony", "czarny",
"brązowy");
        foreach (string kolor in k)
        {
            Console.WriteLine(kolor);
        }
        Console.ReadKey();
    }
}
```

Wskazówki

- W programie należy użyć interfejsu **IEnumerable<T>**. Interfejs ten zawiera jedną metodę `GetEnumerator()`.
- W tym i w wielu kolejnych zadaniach będę oczekiwać od studentów większej samodzielności w poszukiwaniu informacji o konstrukcjach potrzebnych do wykonania określonego zadania.
- Zalecam poszukanie w wyszukiwarce frazy zawierającej teksty **IEnumerable<T>** oraz `yield`.
- O tajemniczym słowie `yield`, nazywanym przez niektórych „czarną magią”, można poczytać w podręczniku „C# 6.0 w pigułce”, zob. darmowy rozdział dostępny na stronie Helionu: <https://pdf.helion.pl/c6pig6/c6pig6.pdf> (temat „Implementowanie interfejsów przeliczeniowych” (str. 15 w pliku pdf).
- Wyjaśnienie na temat `params` (występuje w konstruktorze klasy **Kolory**): Podręcznik Kempa A., Staś T, *Wstęp do programowania w C#* (c-sharp.ue.katowice.pl), str. 134 i 135.

