

## Zadanie – agregacja obiektów, interfejsy i serializacja

---

Zadanie polega na utworzeniu klasy agregującej Członków zespołu i wykorzystującej klasę KierownikZespołu oraz napisaniu funkcji, które zarządzają tą agregacją.

---

Otwórz projekt o nazwie **cw2** (z poprzednich zajęć).

1. Klasę **Osoba** oznacz jako abstrakcyjna. Dopisz dwie nowe klasy, dziedziczące po **Osoba**:

**CzlonekZespołu** oraz **KierownikZespołu**

Wykorzystując konstruktor parametryczny klasy **Osoba** dopisz analogiczne konstruktory parametryczne dla nowo utworzonych klas (wykorzystaj konstrukcję : **base**).

Klasa **CzlonekZespołu** ma dodatkowo atrybut:

**string** funkcja – opisująca funkcję członka zespołu

Klasa **KierownikZespołu** ma dodatkowo atrybut:

**int** doswiadczenie – zawierającą liczbę lat przepracowanych na stanowisku kierownika.

Przesłoń metody **string ToString()**, tak żeby uwzględniała nowe atrybuty.

W programie **Main()** przetestuj klasę **CzlonekZespołu** dla następujących osób:

Beata Nowak 1992-10-22 92102201347 K projektant

Jan Janowski 1992-03-15 92031507772 M Jankowski programista

Przetestuj klasę **KierownikZespołu** dla następującej osoby:

Adam Kowalski 1990-07-01 90070100211 M 5

2. Napisz klasę **Zespol**, która zawiera następujące zmienne:

```
private string nazwa;  
private KierownikZespołu kierownik;  
private List<CzlonekZespołu> czlonkowie;
```

Napisz konstruktory:

**Zespol()** – który inicjalizuje nazwę i kierownika na *null*, a listę czlonkowie na pustą listę

**Zespol(string nazwa, KierownikZespołu kierownik)** – który wywołuje konstruktor domyślny, a następnie ustawia nazwę oraz kierownika zespołu na podane wartości

Napisz (lub wygeneruj) metody akcesorowe dla zmiennych **nazwa** oraz **kierownik**.

3. Dopisz publiczne metody:

**void DodajCzlonka(CzlonekZespołu c)** – która dodaje członka zespołu do listy **czlonkowie**.

**string ToString()** – przesłania metodę klasy **Object**, tak by wypisywała nazwę zespołu, kierownika, a następnie członków zespołu (możesz wykorzystać klasę **StringBuilder**).

Przetestuj w metodzie **Main()** klasę **Zespol** poprzez utworzenie kierownika zespołu, a następnie utworzenie i dodanie członków zespołu do **Zespołu** i w końcu wypisanie zawartości tego **Zespołu**.

#### Przykładowe dane:

---

Zespół : Grupa IT

Kierownik : Adam Kowalski 01.07.1990 90070142412 M 5

Witold Adamski 22.10.1992 92102266738 M sekretarz

Jan Janowski 15.03.1992 92031532652 M programista

Jan But 16.05.1992 92051613915 M programista

Beata Nowak 22.11.1993 93112225023 K projektant

Anna Mysza 22.07.1991 91072235964 K projektant

---

#### 4. Dopisz publiczne metody:

`bool JestCzlonkiem(string PESEL)` – która zwraca prawdę, jeżeli w tablicy **czlonkowie** znajduje się osoba o podanym numerze PESEL, fałsz w przeciwnym wypadku.

`bool JestCzlonkiem(string imie, string nazwisko)` – która zwraca prawdę, jeżeli w tablicy **czlonkowie** znajduje się osoba o podanym imieniu i nazwisku, fałsz w przeciwnym wypadku.

`void UsunCzlonka(string PESEL)` – która usuwa (jeśli jest) członka zespołu o podanym numerze PESEL z tablicy **czlonkowie**.

`void UsunCzlonka(string imie, string nazwisko)` – która usuwa (jeśli jest) członka zespołu o podanym imieniu i nazwisku z tablicy **czlonkowie**.

`void UsunWszystkich()` – która usuwa wszystkich członków zespołu.

`List<CzlonekZespołu> WyszukajFunkcje(string funkcja)` – która zwraca listę wszystkich członków pełniących podaną funkcję.(można użyć: `FindAll()`)