

Wprowadzenie

Celem zadania jest stworzenie prostej aplikacji do zarządzania kontaktami.

1. Uruchom środowisko Visual Studio, a następnie stwórz nowy projekt aplikację konsolową w języku C# (*File -> New -> Project, Console Application*).
2. Dodaj do projektu nową klasę reprezentującą osobę. W tym celu kliknij prawym przyciskiem myszy (PPM) na projekcie w oknie Solution Explorer i wybierz *Add -> New Item -> Class*. Nazwij plik *Osoba.cs* i zatwierdź wybór.



Zwróć uwagę, że w pliku automatycznie została utworzona klasa *Osoba*. Jest to konwencja, której dla czytelności kodu warto się trzymać – każda klasa przechowywana w osobnym pliku o takiej samej nazwie.

3. Dodaj do klasy osoba pole imię typu *string*.
4. W głównej funkcji programu zadeklaruj zmienną typu *Osoba*, a następnie przypisz do niej nowy obiekt tej samej klasy. Spróbuj przypisać utworzonej osobie swoje imię, a następnie zbuduj projekt (Ctrl + Shift + B). Czy kompilacja się powiodła?
5. Odczytaj komunikat błędu. Jaki jest powód zaistniałej sytuacji?
6. Dodaj do pola imię odpowiedni modyfikator dostępu. Przy okazji, zmień nazwę pola tak, aby zaczynała się od wielkiej litery.



To kolejna konwencja nazewnictwa: małymi literami nazywamy pola i metody prywatne, a wielkimi – publiczne.

7. W funkcji głównej spróbuj ponownie ustawić imię utworzonej osobie.
8. Dodaj w funkcji głównej instrukcję wypisującą podane imię na ekran i uruchom program (Ctrl + F5).
9. Dodaj do klasy pole umożliwiające wprowadzenie nazwiska, następnie przypisz utworzonej osobie swoje nazwisko i wypisz je na ekran razem z imieniem w następującym formacie:

Nazywam się <imię> <nazwisko>.



Odwołania do pól i metod wewnątrz klasy zawsze poprzedzamy odwołaniem do zmiennej *this*, np. *this.Nazwisko*.

10. Dodaj w klasie `Osoba` metodę `PrzedstawSie()`, zwracającą powyższy tekst, a następnie skorzystaj z niej w funkcji głównej programu w celu wypisania danych swojej osoby.
11. Dodaj w klasie `Osoba` metodę `UstawDane()`, która przyjmuje jako parametry imię i nazwisko i ustawia wartości odpowiadających im pól. Skorzystaj z metody w funkcji głównej programu.



Kolejna konwencja nazewnictwa mówi aby nazwy parametrów rozpoczynały się małymi literami.

12. Zamień pola na właściwości w taki sposób, aby ustawianie imienia i nazwiska było możliwe jedynie wewnątrz klasy, a odczyt w dowolnym kontekście.
13. Dodaj do projektu nową klasę reprezentującą adres. Klasa ma umożliwiać przechowywanie ulicy, numeru domu, numeru mieszkania, kodu pocztowego, miasta oraz państwa. Ponadto, powinna posiadać właściwość tylko do odczytu umożliwiającą odczyt adresu w formacie „pocztowym”, tj.:

Ul. Nazwa Ulicy 11/1
11-111 Miasto
Państwo

14. Dodaj w klasie `Osoba` właściwość `Adres` przechowującą adres zamieszkania danej osoby.
15. Dostosuj funkcje `PrzedstawSie()` i `UstawDane()` do nowej postaci klasy.
16. Dodaj w klasie `Osoba` nową właściwość tylko do odczytu umożliwiającą odczyt pełnego adresu korespondencyjnego. Posłuż się utworzoną wcześniej właściwością z klasy `Adres`.
17. Dostosuj kod w głównej funkcji programu do nowej definicji klasy `Osoba`.
18. Zmodyfikuj aplikację konsolową aby działała jak książka adresowa. Aplikacja po uruchomieniu prosi użytkownika o podanie liczby kontaktów do wprowadzenia, a następnie prosi o uzupełnienie danych kolejnych osób. Po wprowadzeniu wszystkich danych, wyświetlana jest lista wszystkich kontaktów w formacie:

N. <imię> <nazwisko>

gdzie N to liczba porządkowa. W celu wyświetlenia dokładnych danych korespondencyjnych użytkownik musi wpisać liczbę odpowiadającą danemu kontaktowi i zatwierdzić wybór klawiszem Enter. Ponowne wciśnięcie klawisza Enter powoduje powrót do listy wszystkich kontaktów.