

Ćwiczenie 1

Temat: Klasa + obiekt

Utwórz klasę `Person`, która posiada:

- pole `Name`
- pole `Age`
- metodę `Introduce()`, która wypisuje:

Cześć, mam na imię Jan i mam 20 lat.

W `Main()`:

- utwórz obiekt
 - przypisanie wartości
wywołaj metodę
-

Ćwiczenie 2

Temat: Konstruktor

Zmodyfikuj klasę `Person`, aby:

- nie można było tworzyć osoby bez podania imienia i wieku
- konstruktor przyjmował dwa parametry

Przykład użycia:

```
Person p = new Person("Anna", 25);
```

Ćwiczenie 3

Temat: `private` + `property`

Zrób klasę `BankAccount`, która posiada:

- prywatne pole `balance`
- property `Balance`
- nie pozwala ustawić wartości ujemnej

Dodaj metodę `Deposit(double amount)`, która zwiększa saldo.

Przetestuj w `Main()`.

Ćwiczenie 4

Temat: Klasa bazowa + potomna

Utwórz klasę `Vehicle` z:

- property `Brand`
- metodą `Start()`, która wypisuje „Pojazd uruchomiony”

Utwórz klasę `Car`, która dziedziczy po `Vehicle`.

W `Main()`:

- utwórz obiekt `Car`
 - wywołaj `Start()`
-

Ćwiczenie 5

Temat: Nadpisywanie metod

Zmodyfikuj `Vehicle`:

- metoda `Start()` ma być `virtual`
- w klasie `Car` nadpisz metodę `Start()`, aby wypisywała:

Samochód odpala silnik

Sprawdź polimorfizm:

```
Vehicle v = new Car();  
v.Start();
```

Ćwiczenie 6

Temat: Wywołanie konstruktora klasy bazowej

Zrób klasę `Employee`, która dziedziczy po `Person`.

- `Person` ma konstruktor (`name`, `age`)
- `Employee` ma dodatkowe pole `Salary`
- użyj `: base(name, age)`

Dodaj metodę `ShowInfo()`.

Ćwiczenie 7

Temat: Struct vs Class

Utwórz `struct Rectangle` z:

- polami `Width`, `Height`
- metodą `Area()`, która zwraca pole powierzchni

W `Main()`:

- utwórz dwa obiekty
- przypisz jeden do drugiego
- zmień wartość w drugim
- sprawdź czy pierwszy się zmienił