Tworzenie\_instancji\_i\_uzywanie\_metod

Kasper Dobiech

# 1. Wprowadzenie

W języku C# programowanie opiera się w dużej mierze na **programowaniu obiektowym**. Podstawowym elementem tego podejścia są **klasy** i tworzone na ich podstawie **instancje**, czyli **obiekty**. Aby móc korzystać z metod (czyli funkcji przypisanych do klasy), najpierw musimy utworzyć obiekt danej klasy.

Przypomnienie:

**Klasa** to wzór lub „przepis”, który opisuje, jakie dane (pola) oraz jakie zachowania (metody) będą mieć obiekty tworzone na jej podstawie.

Obraz zawierający tekst, zrzut ekranu, wyświetlacz, oprogramowanie

Zawartość wygenerowana przez AI może być niepoprawna.

**Instancja** to konkretny egzemplarz klasy — coś, co tworzymy na podstawie definicji klasy.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Czcionka

Zawartość wygenerowana przez AI może być niepoprawna.

# 2. Teoria

**Co to jest tworzenie instancji?**

**Tworzenie instancji** (ang. *instantiation*) to proces **tworzenia konkretnego obiektu na podstawie klasy**.

* **Klasa** = przepis / szablon (np. plan samochodu)
* **Instancja (obiekt)** = gotowy egzemplarz (np. Twój konkretny samochód)

Do utworzenia obiektu używamy słowa kluczowego **new**.

Samochod auto = new Samochod();

Rozbijmy to:

|  |  |
| --- | --- |
| Element | Znaczenie |
| Samochod | Typ danych → nazwa klasy |
| auto | Nazwa zmiennej → nasz obiekt |
| new Samochod() | Utworzenie instancji za pomocą konstruktora |

* new rezerwuje miejsce w pamięci dla obiektu.
* Samochod() uruchamia konstruktor (czyli specjalną metodę inicjującą obiekt).
* Zmienna auto otrzymuje referencję do nowego obiektu.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, wyświetlacz

Zawartość wygenerowana przez AI może być niepoprawna.

Tworzymy wiele instancji tej samej klasy

Każdy obiekt jest **oddzielny i niezależny**, mimo że powstał z tej samej klasy.

**Kiedy tworzymy instancje?**

* Gdy chcemy **korzystać z metod i pól klasy**
* Gdy potrzebujemy **wiele podobnych obiektów** (np. wielu uczniów, samochodów, produktów)
* W **programach okienkowych, grach, systemach baz danych** — praktycznie zawsze!

## Co to są metody klasy?

**Metody** to **zachowania** obiektu — czyli czynności, które obiekt może wykonać.

Jeśli **klasa** to przepis na coś, a **instancja** to gotowy przedmiot, to **metody** są **czynnościami**, jakie ten obiekt potrafi robić.

**Jak wygląda metoda w C#?**

Metoda to fragment kodu zapisany wewnątrz klasy. Ma:

* **Typ zwracany** (void jeśli nic nie zwraca)
* **Nazwę**
* **Opcjonalne parametry**
* **Ciało**, czyli kod, który zostanie wykonany

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Zawartość wygenerowana przez AI może być niepoprawna.

**Jak wywołać metodę?**

Najpierw tworzymy **instancję**, potem używamy **kropki**:

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Zawartość wygenerowana przez AI może być niepoprawna.

Metody mogą przyjmować parametry

public void UstawPredkosc(int kmh)

{

Console.WriteLine("Ustawiono prędkość na " + kmh + " km/h");

}

Wywołanie: auto.UstawPredkosc(60);

Metody mogą coś zwracać:

public int PobierzRokProdukcji()

{

return 2010;

}

Wywołanie: int rok = auto.PobierzRokProdukcji();

Rodzaje metod (w skrócie)

|  |  |  |
| --- | --- | --- |
| Rodzaj | Opis | Przykład |
| Instancyjne | Wywoływane na konkretnym obiekcie | auto.UruchomSilnik() |
| Statyczne | Należą do klasy, nie do obiektu | Math.Sqrt(16) |

# 4. Podsumowanie

**Tworzenie instancji i używanie metod w C#:**

* **Klasa** to szablon opisujący dane i zachowania obiektu.
* **Instancja (obiekt)** to konkretny egzemplarz klasy, tworzony za pomocą new.
* **Metody** to funkcje klasy, które wykonują określone czynności i wywołuje się je na obiektach przez kropkę: obiekt.Metoda().