Enkapsulacja private/ public

Kasper Dobiech

# 1. Wprowadzenie

W programowaniu obiektowym **enkapsulacja** oznacza **ukrywanie szczegółów wewnętrznych obiektu** i kontrolowanie, jak inni programiści mogą korzystać z jego danych.

* Dzięki temu **chronimy pola klasy przed przypadkową zmianą**.
* Udostępniamy dane tylko poprzez **bezpieczne metody lub właściwości**.
* W C# do tego celu używamy modyfikatorów dostępu: **private** (ukrywa) i **public** (udostępnia)

# 2. Teoria

**Enkapsulacja** (ang. *encapsulation*) to zasada programowania obiektowego, która polega na **ukrywaniu danych obiektu** i kontrolowaniu dostępu do nich.

* Chronimy pola klasy przed bezpośrednią modyfikacją.
* Umożliwiamy dostęp do danych tylko w sposób **bezpieczny i kontrolowany**.

|  |  |  |
| --- | --- | --- |
| private | Pole/metoda dostępna tylko wewnątrz klasy | private int wiek; |
| public | Pole/metoda dostępna z każdej części programu | public void WyswietlInfo() { ... } |

Zasada: **pola klasy zazwyczaj ustawiamy jako private**, a dostęp do nich zapewniamy przez **metody lub właściwości public**.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Zawartość wygenerowana przez AI może być niepoprawna.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Zawartość wygenerowana przez AI może być niepoprawna.

* Nie możemy zrobić jan.wiek = -5; bezpośrednio, bo pole wiek jest private.
* Wszystkie zmiany przechodzą przez **kontrolowane metody**.

**Dlaczego to jest przydatne?**

* Chroni przed **błędnymi danymi**.
* Pozwala **zmieniać implementację wewnątrz klasy** bez wpływu na resztę programu.
* Sprawia, że kod jest **bezpieczniejszy i bardziej czytelny**.

**Co to jest właściwość?**

**Właściwość (property)** to specjalny sposób udostępniania **prywatnych pól klasy** w sposób kontrolowany.

* Pozwala **odczytywać (get)** i **ustawiać (set)** wartości.
* Dzięki temu nie musimy pisać osobnych metod do odczytu i zapisu.

Obraz zawierający tekst, zrzut ekranu, wyświetlacz, oprogramowanie

Zawartość wygenerowana przez AI może być niepoprawna.

* get → zwraca wartość pola (return wiek)
* set → przypisuje wartość do pola (wiek = value)
* value → specjalna zmienna w set, reprezentuje wartość przypisywaną do właściwości

Jak korzystać z właściwości?

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, wyświetlacz

Zawartość wygenerowana przez AI może być niepoprawna.

* Nie można ustawić wieku na ujemny – kontrola jest w set.
* Dzięki temu **dane pozostają bezpieczne**, a kod jest czytelniejszy.

# Podsumowanie

Enkapsulacja = ukrywanie danych i kontrola dostępu.

private = dostęp tylko wewnątrz klasy.

public = dostęp z każdego miejsca w programie.

Pola klasy najlepiej ustawiać jako private, a dostęp zapewniać przez publiczne metody lub właściwości.

Dzięki enkapsulacji program staje się bezpieczniejszy, bardziej czytelny i łatwiejszy w utrzymaniu.

|  |  |
| --- | --- |
| get | Odczytuje wartość pola |

|  |  |
| --- | --- |
| set | Ustawia wartość pola, z możliwością kontroli |

|  |  |
| --- | --- |
| value | Reprezentuje przypisywaną wartość w set |