Enkapsulacja private/ public

Kasper Dobiech

# 1. Cel ćwiczenia

Zapoznanie ucznia z enkapsujacją

# 2. Wymagania wstępne

Środowisko Visual Studio

Zapoznanie z częścią teoretyczną

# 3. Instrukcja krok po kroku

**Zadanie 1: Klasa „Telefon”**

1. Stwórz klasę Telefon z prywatnymi polami:
   * marka (string)
   * bateria (int, 0–100)
2. Utwórz właściwości:
   * Marka – dowolna wartość
   * Bateria – tylko wartości między 0 a 100
3. Dodaj metodę WyswietlStan(), która pokazuje markę telefonu i poziom baterii.
4. W Main():
   * Stwórz obiekt telefonu.
   * Przypisz wartości i spróbuj ustawić Bateria = 150 – sprawdź, czy kontrola działa.
   * Wywołaj metodę WyswietlStan().

**Zadanie 2: Klasa „Pies”**

1. Stwórz klasę Pies z prywatnymi polami:
   * imie (string)
   * wiek (int)
2. Utwórz właściwości Imie i Wiek z kontrolą, aby wiek był większy od 0.
3. Dodaj metodę Szczekaj(), która wypisuje "Hau! Hau!".
4. W Main():
   * Stwórz obiekt psa.
   * Ustaw właściwości i wywołaj metodę Szczekaj().
   * Spróbuj ustawić nieprawidłowy wiek i sprawdź komunikat.

**Zadanie 3: Klasa „KontoBankowe”**

1. Stwórz klasę KontoBankowe z prywatnym polem saldo (decimal).
2. Utwórz właściwość Saldo z **tylko get** (nie pozwalamy ustawiać salda z zewnątrz).
3. Dodaj metody:
   * Wplac(decimal kwota) – zwiększa saldo
   * Wyplac(decimal kwota) – zmniejsza saldo, ale nie pozwala na ujemne wartości
4. W Main():
   * Stwórz konto i przetestuj wpłaty i wypłaty.

Zadanie 4:  **Klasa „GraRPG”**

Stwórz klasę reprezentującą postać w grze RPG z kontrolowanymi statystykami i metodami do interakcji.

**Wymagania klasy Postac**

* Prywatne pola:
  + imie (string)
  + zycie (int, max 100)
  + mana (int, max 50)
  + poziom (int, >=1)
* Właściwości:
  + Imie – nie może być puste
  + Zycie – kontrola 0–100
  + Mana – kontrola 0–50
  + Poziom – kontrola >=1
* Metody:
  + Atak() – zmniejsza mana o 10 (jeśli wystarczy) i wypisuje komunikat, np. "Postać X atakuje!"
  + Leczenie() – zwiększa zycie o 20, ale nie przekracza 100
  + Awansuj() – zwiększa poziom o 1 i wyświetla komunikat o awansie

**Wymagania w Main()**

1. Stwórz kilka postaci.
2. Przypisz im właściwości, próbując też ustawić nieprawidłowe wartości (np. ujemne życie, zbyt dużą manę).
3. Wywołaj różne metody (Atak, Leczenie, Awansuj) i sprawdź, czy kontrola w właściwościach działa poprawnie.
4. Wyświetl wszystkie informacje o postaciach po kilku akcjach.