

ĆWICZENIE 1: System rezerwacji bibliotecznej

Cel:

Stwórz system zarządzania zasobami bibliotecznymi z różnymi typami mediów, użytkownikami i zasadami wypożyczeń.

Struktura:

1. Klasa abstrakcyjna Pozycja

- Pola: tytuł (String), id (int), czyDostepna (boolean) – wszystkie private
- Konstruktor z this
- Metoda abstrakcyjna void wyswietlInfo()
- Metoda final boolean czyMoznaWypozyczyc() – zwraca stan dostępności
- Gettery/settery z walidacją (np. ID > 0)

2. Klasy dziedziczące: Ksiazka, Czasopismo, Audiobook

- Każda ma dodatkowe pola:
 - Ksiazka: autor, liczbaStron
 - Czasopismo: numerWydania, issn
 - Audiobook: dlugoscMinuty, lektor
- Każda nadpisuje wyswietlInfo() z własnym formatem

3. Interfejs Rezerwowalny

- Metody: boolean zarezerwuj(), boolean anulujRezerwacje()

4. Klasa Uzytkownik

- Pola: imie, nazwisko, idUzytkownika
- Metoda: void wypożycz(Pozycja p) – zmienia status pozycji

5. Klasa Biblioteka (singleton z final)

- Pole static final MAX_WYPOZYCZEN = 5
- Mapowanie użytkowników do listy wypożyczonych pozycji
- Metoda sprawdzająca limit wypożyczeń

ĆWICZENIE 2: System zarządzania projektami IT

Cel:

Stwórz system do śledzenia projektów programistycznych z różnymi typami zadań, pracownikami i statusami.

Struktura:

1. Klasa abstrakcyjna Zadanie

- Pola: nazwa (String), priorytet (int), status (enum: PLANOWANE, W_TRAKCIE, ZAKONCZONE)
- Konstruktor z walidacją priorytetu (1-5) przy użyciu this
- Metoda abstrakcyjna double obliczSzacowanyCzas()
- Metoda final void zmienStatus(Status nowyStatus)

2. Klasy dziedziczące: BugFix, NowaFunkcjonalnosc, Refaktoryzacja

- BugFix: poziomTrudnosci, czyKrytyczny
- NowaFunkcjonalnosc: liczbaPointow, technologia
- Refaktoryzacja: czyWymagaTestow, staryKod

3. Interfejs Przypisywalny

- Metody: boolean przypiszPracownika(Pracownik p), boolean usunPrzypisanie()

4. Klasa Pracownik (final)

- Pola: imie, nazwisko, stanowisko, maxZadan (static final = 3)
- Metoda: boolean czyMozePodjacZadanie()

5. Klasa Projekt

- Lista zadań z polimorficznym przetwarzaniem
- Metoda void raportujPostep() – dla każdego zadania wywołuje obliczSzacowanyCzas()