

Zadanie 1

a) Stwórz publiczną klasę `Osadnik`. Dodaj do klasy pola:

- `string nazwisko`
- `string wioska`
- `DateTime dataNarozdin`
- `decimal masa`
- `double predkosc`

Dodaj do klasy statyczne pole:

- `int populacja`

Dodaj do klasy właściwości dla wszystkich pól.

b) Dodaj do klasy konstruktor statyczny, który inicjuje pole `populacja` wartością 15.

c) Dodaj do klasy konstruktor nieparametryczny, zwiększający `populacja` o 1.

d) Dodaj konstruktor parametryczny, inicjujący wszystkie pola, wykorzystując konstruktor nieparametryczny.

e) Dodaj publiczną metodę wirtualną `double Sila()`, zwracającą iloczyn `masa` i `predkosc`.

f) Przesłoń metodę `ToString()`, aby wypisywała:
"Kazimierz z wioski Gniezno, Siła: 1000".

Zadanie 2

a) Stwórz publiczną enumerację `Stopien` z wartościami: `Strzelec`, `Pikinier`, `Kawalerzysta`.

b) Stwórz publiczną klasę `Zolnierz`, dziedziczącą po `Osadnik`. Dodaj:

- `string identyfikator`
- `Stopien stopien`

Dodaj właściwości dla tych pól, z prywatną metodą `set` dla `stopien`.

c) Dodaj konstruktor nieparametryczny, inicjujący `stopien` jako `Pikinier` i tworzący `identyfikator` według schematu: "PIK-021" (na podstawie `populacja`).

d) Przesłoń metodę `Sila()`, dodając wartości: 15 dla `Pikinier`, 25 dla `Strzelec`, 60 dla `Kawalerzysta`.

e) Przesłoń metodę `ToString()`, aby wypisywała
"Kazimierz z miasta Gniezno, Strzelec, STR-021, Siła: 1025".

Zadanie 3

a) Utwórz klasę `Dowodca`, dziedziczącą po `Osadnik`, z polem:

- `string rang`

Dodaj właściwość dla tego pola i odpowiedni konstruktor parametryczny.

b) Stwórz klasę błędu `NieprawidlowaRangaException`, dziedziczącą po `Exception`.

c) Przy ustawianiu `Rang`, rzuć `NieprawidlowaRangaException`, jeśli nie zaczyna się od dużej litery lub ma mniej niż 4 znaki.

d) Przesłoń metodę `Siła()`, mnożąc wartość bazową przez 100.

e) Przesłoń `ToString()`, formatując:
"Hetman z wioski Gniezno – Tadeusz".

Zadanie 4:

a) Utwórz klasę `Brygada`, zawierającą pola:

- `string nazwa`
- `General general`
- `List<Żołnierz> żołnierze`

b) Stwórz konstruktor nieparametryczny, który inicjuje `żołnierze` jako nową, pustą listę oraz `general` jako nową instancję klasy `General`.

c) Utwórz metodę `string Skład()`, zwracającą informacje o składzie oddziału w formacie: "Artylerzysta: 3, Piechur: 4, Kuznik: 2".

d) Stwórz metodę `double SiłaOddziału()`, zwracającą sumę wartości `Siła()` dla wszystkich obiektów na liście `żołnierze` oraz dodatkowo `Siła()` generała.

e) Dodaj metodę `void Rekrutuj(Żołnierz)`, która dołączy wskazanego żołnierza do oddziału.

f) Dodaj metodę `void Rozwiaz(Żołnierz)`, która usunie wskazanego żołnierza z oddziału.

Zadanie 5:

Dodaj do klasy `Brygada` dwie publiczne statyczne metody:

- `void ZapiszDoPliku(Brygada, String)` - zapisująca przekazaną brygadę do pliku o wskazanej nazwie w formacie binarnym.
- `Brygada WczytajZPliku(String)` - wczytująca brygadę z pliku o wskazanej nazwie i zwracająca ją.

Zadanie 6:

a) Zadeklaruj, że klasa `Zolnierz` implementuje interfejs `Comparable<Zolnierz>`. Zaimplementuj metody interfejsu tak, aby domyślnie porównywać żołnierzy na podstawie wartości metody `Sila()`. W przypadku równej `Sila()`, porządkuj według rangi: Kuznik > Artylerzysta > Piechur.

b) Dodaj do klasy `Brygada` publiczną metodę `void Sortuj()`, która posortuje żołnierzy z listy `zolnierze` według zaimplementowanego komparatora.