

Wariant 507

Wykonaj poniższe czynności:

- Utwórz pakiet `com.electronics.store` i umieść w nim oddzielne pliki dla klas.
- Zaprojektuj klasę `Appliance` z prywatnymi polami: `type` (typu `String`) oraz `energyConsumption` (typu `double`).
- Zaimplementuj konstruktor, który otrzymuje dwa argumenty odpowiadające polom klasy i mające nazwy identyczne jak pola.
- Stwórz publiczne metody dostępne (getter i setter) dla obu pól.
- Ustal mechanizm walidacji dla pola `energyConsumption`, aby upewnić się, że jego wartość jest nieujemna. W przypadku przekazania wartości ujemnej w konstruktorze, ustaw `energyConsumption` na wartość domyślną 0.5. Setter dla tego pola nie powinien akceptować wartości ujemnych i nie zmieniać wartości pola w takim wypadku.
- Stwórz klasę `TestAppliance` z metodą `main`. W głównej metodzie utwórz instancję `Appliance` i przetestuj działanie wszystkich metod dostępnych.

Zadanie należy umieścić we własnym repozytorium na Githubie.

Wariant 508

Wykonaj poniższe czynności:

- Utwórz pakiet `com.urban.planning` i w oddzielnych plikach umieść dwie klasy.
- Zaprojektuj klasę `Park` z prywatnymi polami: `name` (typu `String`) oraz `area` (typu `double`) reprezentującymi nazwę parku i jego powierzchnię w hektarach.
- Zaimplementuj konstruktor, który przyjmuje dwa argumenty odpowiadające nazwie i powierzchni, posiadające takie same nazwy jak pola klasy.
- Utwórz publiczne metody dostępne (getter i setter) dla obu pól klasy.
- Dodaj walidację dla pola `area`, aby upewnić się, że jest dodatnie. Jeśli w konstruktorze zostanie podana wartość ujemna, ustaw powierzchnię parku na wartość domyślną 2.5. Setter dla `area` nie powinien akceptować wartości ujemnych i nie zmieniać wartości pola w przypadku podania takiej wartości.
- Stwórz klasę `TestPark` z metodą `main`, w której utworzysz instancję klasy `Park` i wykorzystasz wszystkie metody dostępne co najmniej raz.

Zadanie należy umieścić we własnym repozytorium na Githubie.

Wariant 509

Wykonaj poniższe czynności:

- Utwórz pakiet `com.fitness.tracker` i umieść w nim w oddzielnych plikach dwie klasy.
- Zaprojektuj klasę `WorkoutSession` z prywatnymi polami: `activityName` (typu `String`) oraz `durationMinutes` (typu `int`) do przechowywania nazwy aktywności i czasu trwania sesji treningowej w minutach.
- Zaimplementuj konstruktor, który przyjmuje dwa argumenty odpowiadające polom klasy i mające identyczne nazwy jak te pola.
- Stwórz publiczne metody dostępne (getter i setter) dla obu pól klasy.
- Dodaj walidację dla pola `durationMinutes`, by upewnić się, że czas trwania nie jest mniejszy niż 1 minuta. W przypadku podania wartości mniejszej niż 1 w konstruktorze, ustaw czas trwania sesji na 30 minut. Setter dla `durationMinutes` nie powinien akceptować wartości mniejszych niż 1 i powinien zostawiać pole niezmienione w takim przypadku.
- Stwórz klasę `TestWorkoutSession` z metodą `main`, w której stworzysz instancję `WorkoutSession` i przetestujesz działanie wszystkich metod dostępnych.

Zadanie należy umieścić we własnym repozytorium na Githubie.