

Wariant 251

- Utwórz pakiet `library` dla realizacji poniższych zadań.
- Napisz klasę `Book`, która zawiera pola: `title` (typu `String`), `author` (typu `String`) i `yearPublished` (typu `int`). Zaimplementuj interfejs `Comparable` w taki sposób, aby obiekty klasy `Book` były sortowane malejąco według roku wydania.
- W klasie `TestBook` stwórz tablicę 5 obiektów klasy i posortuj ją według sprecyzowanego kryterium.

Zadanie należy umieścić we własnym repozytorium na Githubie.

Wariant 252

- Utwórz pakiet `transport` dla zarządzania pojazdami.
- Napisz klasę `Vehicle`, która zawiera pola: `model` (typu `String`), `speed` (typu `double`) i `fuelConsumption` (typu `double`). Zaimplementuj interfejs `Comparable` tak, aby obiekty klasy `Vehicle` były sortowane rosnąco według zużycia paliwa.
- W klasie `TestVehicle` stwórz tablicę 4 obiektów klasy `Vehicle` i posortuj ją według sprecyzowanego kryterium.

Zadanie należy umieścić we własnym repozytorium na Githubie.

Wariant 253

- Utwórz pakiet `inventory` przeznaczony do zarządzania zasobami.
- Stwórz klasę `Product`, która zawiera pola: `id` (typu `int`), `quantity` (typu `int`) i `unitPrice` (typu `double`). Zaimplementuj interfejs `Comparable`, aby obiekty klasy `Product` były sortowane rosnąco według ceny jednostkowej.
- W klasie `TestProduct` utwórz listę 5 obiektów klasy `Product` i posortuj ją według wybranego kryterium.

Zadanie należy umieścić we własnym repozytorium na Githubie.

Wariant 254

- Stwórz pakiet `gadget` do zarządzania urządzeniami elektronicznymi.
- Napisz klasę `Gadget`, która zawiera pola: `type` (typu `String`), `batteryLife` (typu `int`) i `price` (typu `double`). Zaimplementuj interfejs `Comparable`, aby obiekty klasy `Gadget` były sortowane malejąco według długości życia baterii.
- W klasie `TestGadget` utwórz zestaw 4 obiektów klasy `Gadget` i posortuj je według wybranego kryterium.

Zadanie należy umieścić we własnym repozytorium na Githubie.