

Wariant 607

- Stwórz pakiet `pl.edu.uwm.vehicles` i umieść w nim oddzielne pliki dla poniższych klas.
- Stwórz klasę `Car` z prywatnymi polami: `model` (typu `String`, pol. `model`), `yearOfManufacture` (typu `int`, pol. rok produkcji) oraz `mileage` (typu `double`, pol. przebieg).
- Zaimplementuj konstruktor parametryczny ustawiający pola klasy z argumentów.
- Nadpisz metodę `toString()`, tak aby zwracała napis postaci:
Samochód. Model: [model], rok produkcji: [yearOfManufacture], przebieg: [mileage] km.
Zwróć uwagę na wielkość znaków i znaki interpunkcyjne.
- Stwórz klasę `TestCar` z metodą `main`. W metodzie `main` stwórz obiekt klasy `Car` i przetestuj wszystkie metody.

Zadanie należy umieścić we własnym repozytorium na Githubie.

Wariant 608

Wykonaj poniższe czynności:

- Stwórz pakiet `pl.edu.uwm.events` i umieść w nim oddzielne pliki dla poniższych klas.
- Stwórz klasę `Event` z prywatnymi polami: `name` (typu `String`, pol. nazwa), `date` (typu `String`, pol. data) oraz `location` (typu `String`, pol. lokalizacja).
- Zaimplementuj konstruktor parametryczny ustawiający pola klasy z argumentów.
- Nadpisz metodę `toString()`, tak aby zwracała napis postaci:
Wydarzenie. Nazwa: [name], data: [date], lokalizacja: [location].
Zwróć uwagę na wielkość znaków i znaki interpunkcyjne.
- Stwórz klasę `TestEvent` z metodą `main`. W metodzie `main` stwórz obiekt klasy `Event` i przetestuj wszystkie metody.

Zadanie należy umieścić we własnym repozytorium na Githubie.

Wariant 609

Wykonaj poniższe czynności:

- Stwórz pakiet `pl.edu.uwm.restaurant` i umieść w nim oddzielne pliki dla poniższych klas.
- Stwórz klasę `MenuItem` z prywatnymi polami: `name` (typu `String`, pol. nazwa), `ingredients` (typu `String`, pol. składniki) oraz `price` (typu `double`, pol. cena).
- Zaimplementuj konstruktor parametryczny ustawiający pola klasy z argumentów.
- Nadpisz metodę `toString()`, tak aby zwracała napis postaci:
Pozycja menu. Nazwa: [name], składniki: [ingredients], cena: [price] zł.
Zwróć uwagę na wielkość znaków i znaki interpunkcyjne.
- Stwórz klasę `TestMenuItem` z metodą `main`. W metodzie `main` stwórz obiekt klasy `MenuItem` i przetestuj wszystkie metody.

Zadanie należy umieścić we własnym repozytorium na Githubie.

Wariant 610

- Stwórz pakiet `pl.edu.uwm.akt6` i umieść w nim oddzielne pliki dla poniższych klas.
- Stwórz klasę `Car` z prywatnymi polami: `model` (typu `String`), `yearManufactured` (typu `int`) oraz `price` (typu `double`).
- Zaimplementuj konstruktor parametryczny ustawiający pola w klasie z argumentów.
- Nadpisz metodę `toString()` tak aby zwracała napis postaci:
`Samochód. Model: [model], rok produkcji [yearManufactured], cena: [price].`
- Stwórz klasę `TestCar` z metodą `main`. W metodzie `main` stwórz obiekt klasy `Car` i przetestuj wszystkie metody.

Zadanie należy umieścić we własnym repozytorium na Githubie.

Wariant 611

Wykonaj poniższe czynności:

- Stwórz pakiet `pl.edu.uwm.akt6` i umieść w nim oddzielne pliki dla poniższych klas.
- Stwórz klasę `Laptop` z prywatnymi polami: `brand` (typu `String`), `yearReleased` (typu `int`) oraz `price` (typu `double`).
- Zaimplementuj konstruktor parametryczny ustawiający pola w klasie z argumentów.
- Nadpisz metodę `toString()` tak aby zwracała napis postaci:
`Laptop. Marka: [brand], rok wydania [yearReleased], cena: [price].`
- Stwórz klasę `TestLaptop` z metodą `main`. W metodzie `main` stwórz obiekt klasy `Laptop` i przetestuj wszystkie metody.

Zadanie należy umieścić we własnym repozytorium na Githubie.

Wariant 612

Wykonaj poniższe czynności:

- Stwórz pakiet `pl.edu.uwm.akt6` i umieść w nim oddzielne pliki dla poniższych klas.
- Stwórz klasę `Smartphone` z prywatnymi polami: `model` (typu `String`), `yearLaunched` (typu `int`) oraz `price` (typu `double`).
- Zaimplementuj konstruktor parametryczny ustawiający pola w klasie z argumentów.
- Nadpisz metodę `toString()` tak aby zwracała napis postaci:
`Smartphone. Model: [model], rok wprowadzenia [yearLaunched], cena: [price].`
- Stwórz klasę `TestSmartphone` z metodą `main`. W metodzie `main` stwórz obiekt klasy `Smartphone` i przetestuj wszystkie metody.

Zadanie należy umieścić we własnym repozytorium na Githubie.