Kolokwium Programowanie obiektowe 19.01.2022 – Grupa 6

Zadanie 1

Napisz metodę statyczną, która w parametrze przyjmie ciąg znaków *(*typ String). Metoda ma za zadanie odnaleźć (wyświetlić) najdłuższe słowo w podanym ciągu. W przypadku kilku słów o tej samej długości, należy wyświetlić pierwsze z nich. Przetestuj metodę, przekazując do niej **stałą** o wartości:

„Jakie to dziwne, że kiedy człowiek nosi w sercu jakąś własną tajemną teorię, do której właściwie nie chce się przyznać, a usłyszy ją z ust kogoś innego, z pasją jej zaprzecza"

Christie A.

Zadanie 2 – dziedziczenie

Stwórz odpowiedni schemat klas, tak aby były w nim wymodelowane klasy *Vehicle i Car* (patrz niżej), oraz aby **żadne pola/metody nie były w nich zduplikowane**. Pamiętaj o stworzeniu odpowiednich konstruktorów, oraz getterów i setterów. Wykorzystując mechanizm dziedziczenia, pamiętaj o wybraniu odpowiednich modyfikatorów dostępu.

Klasa *Vehicle:*
pola:
- marka
- model
- rok produkcji

metody:
- calculateAge() – zwracającą wiek danego pojazdu (w latach)
- displayInfo() – wyświetlającą informacje o danym pojeździe(np. "This vehicle is: *marka, model*").

Klasa *Car:*
pola:
- marka
- model
- rok produkcji
- zużycie paliwa na 100km
- obecny stan paliwa w litrach
- maksymalny poziom paliwa w litrach

metody:
- calculateAge() – zwracającą wiek danego pojazdu (w latach)
- displayInfo() – wyświetlającą informacje o danym pojeździe(np. "This vehicle is: *marka, model*").

- drive(int distanceInKm) – zmniejszającą ilość paliwa w zależności od parametru *distanceInKm.*
- refuel(int amountOfFuel) – zwiększającą ilość paliwa o podany parametr.

Stwórz następujące obiekty Vehicle oraz Car oraz wyświetl wynik działania wszystkich zaimplementowanych metod (oprócz getterów/setterów).

Vehicle – Marka: Wigry, model: 3, rok produkcji: 1980
Car – Marka: Ford, model: Mustang, rok produkcji: 2014, zużycie paliwa: 11.1, obecny stan paliwa: 20l, maksymalny poziom paliwa: 60l.

Zadanie 3

Stwórz klasę abstrakcyjną *Person,* z następującymi polami: imię, nazwisko, rok urodzenia, adres. Utwórz też wszystkie metody get/set oraz konstruktor. Następnie stwórz metodę printData() wyświetlającą informacje o danym człowieku w następującym formacie:

Name: *imię*
Surname: *nazwisko*
Class: *Nazwa klasy, obiektu dla którego wywołujemy metodę printData().*
Age: *wiek*
Address: *adres*
Do klasy *Person* możesz dodawać dowolne inne metody.
Następnie zdefiniuj następujące klasy dziedziczące po klasie *Person:*

- *Student* – z polami wskazującymi na nazwę uniwersytetu oraz rok, w którym zaczęli studia,
- Employee – z polami wskazującymi na nazwę firmy, w której pracują, oraz rok rozpoczęcia pracy.

W obu klasach nadpisz metodę printData() tak, by wywołały metodę printData() z nad klasy oraz dodatkowo wyświetlały pola dodane w określonej klasie, np:

Name: *imię*
Surname: *nazwisko*
Class: *Nazwa klasy, obiektu dla którego wywołujemy metodę printData().*
Age: *wiek*
Address: *adres*
University: *uniwersytet*
Start year: *startYear*

Utwórz obiekty klas Student i Employee oraz wywołaj dla nich metodę printData()

Do zaimplementowania rozwiązania należy wykorzystać metodę abstrakcyjną.