

Programowanie obiektowe w Javie - klasa, obiekt, konstruktor, modyfikator dostępu

1. Stwórz klasę `Samochod` zawierającą prywatne pola: `marka`, `model`, `rokProdukcji`, `przebieg` oraz `kolor`. Dodaj konstruktor, który przyjmuje wszystkie pola jako argumenty. Dodaj metody dostępne (getter i setter) dla wszystkich pól. Następnie dodaj metodę `wyswietlInformacje()`, która wyświetla wszystkie informacje o samochodzie.
2. Stwórz klasę `Osoba` z prywatnymi polami: `imie`, `nazwisko`, `wiek`, `adres`. Dodaj konstruktor, który przyjmuje wszystkie pola jako argumenty. Dodaj metody dostępne (getter i setter) oraz metodę `przedstawSie()`, która zwraca łańcuch znaków z informacjami o osobie.
3. Stwórz klasę `Ksiazka` z prywatnymi polami: `tytul`, `autor`, `rokWydania`, `wydawnictwo` oraz `liczbaStron`. Dodaj konstruktor, który przyjmuje wszystkie pola jako argumenty. Dodaj metody dostępne (getter i setter) oraz metodę `pokazInformacje()`, która wyświetla informacje o książce.
4. Stwórz klasę `Punkt2D` z prywatnymi polami `x` i `y`, reprezentującymi współrzędne punktu na płaszczyźnie. Dodaj konstruktor, który przyjmuje współrzędne jako argumenty. Dodaj metody dostępne (getter i setter) oraz metodę `odleglosc(Punkt2D innyPunkt)`, która oblicza odległość między dwoma punktami na płaszczyźnie.
5. Stwórz klasę `Prostokat` z prywatnymi polami `szerokosc` i `wysokosc`. Dodaj konstruktor, który przyjmuje długości boków jako argumenty. Dodaj metody dostępne (getter i setter) oraz metody `pole()` i `obwod()`, które obliczają pole powierzchni i obwód prostokąta.
6. Stwórz klasę `Kolo` z prywatnym polem `promien`. Dodaj konstruktor, który przyjmuje promień jako argument. Dodaj metody dostępne (getter i setter) oraz metody `pole()` i `obwod()`, które obliczają pole powierzchni i obwód koła.
7. Stwórz klasę `Student` z prywatnymi polami: `imie`, `nazwisko`, `numerIndeksu`, `rokStudiow` oraz `sredniaOcen`. Dodaj konstruktor, który przyjmuje wszystkie pola jako argumenty. Dodaj metody dostępne (getter i setter) oraz metodę `pokazInformacje()`, która wyświetla informacje o studencie.

8. Stwórz klasę `Pracownik` z prywatnymi polami: `imie`, `nazwisko`, `stanowisko`, `wiek` oraz `placa`. Dodaj konstruktor, który przyjmuje wszystkie pola jako argumenty. Dodaj metody dostępne (getter i setter) oraz metodę `pokazInformacje()`, która wyświetla informacje o pracowniku.
9. Stwórz klasę `KontoBankowe` z prywatnymi polami: `numerKonta`, `wlasciciel`, `saldo` oraz `typKonta`. Dodaj konstruktor, który przyjmuje wszystkie pola jako argumenty. Dodaj metody dostępne (getter i setter) oraz metody `wplac(double kwota)` i `wyplac(double kwota)`, które odpowiednio dodają lub odejmują kwotę od salda konta.
10. Stwórz klasę `Telewizor` z prywatnymi polami: `marka`, `przekatnaEkranu`, `rozdzielczosc`, `czySmartTV` oraz `cena`. Dodaj konstruktor, który przyjmuje wszystkie pola jako argumenty. Dodaj metody dostępne (getter i setter) oraz metodę `pokazInformacje()`, która wyświetla informacje o telewizorze.
11. Stwórz klasę `DziennikOcen` z prywatnymi polami: `imie`, `nazwisko` oraz `oceny` (jako `ArrayList` typu `int`). Dodaj konstruktor, który przyjmuje imię i nazwisko jako argumenty. Dodaj metody dostępne (getter i setter) oraz metody `dodajOcene(int ocena)` i `usunOcene(int indeks)`, które odpowiednio dodają lub usuwają ocenę z listy ocen. Dodaj również metodę `sredniaOcen()` do obliczania średniej ocen.
12. Stwórz klasę `HistoriaTemperatur` z prywatnym polem temperatury (jako `ArrayList` typu `double`). Dodaj konstruktor domyślny. Dodaj metody dostępne (getter i setter) oraz metody `dodajTemperature(double temperatura)` i `usunTemperature(int indeks)`, które odpowiednio dodają lub usuwają temperaturę z listy temperatur. Dodaj również metodę `sredniaTemperatur()` do obliczania średniej temperatur.
13. Stwórz klasę `WynikiTestow` z prywatnymi polami: `imie`, `nazwisko` oraz `wyniki` (jako tablica typu `int`). Dodaj konstruktor, który przyjmuje imię, nazwisko oraz rozmiar tablicy jako argumenty. Dodaj metody dostępne (getter i setter) oraz metodę `dodajWynik(int indeks, int wynik)`, która dodaje wynik testu na podanym indeksie. Dodaj również metodę `sredniWynik()` do obliczania średniego wyniku.
14. Stwórz klasę `ZarzadcaZadan` z prywatnym polem `priorytetyZadan` (jako `ArrayList` typu `int`). Dodaj konstruktor domyślny. Dodaj metody dostępne (getter i setter) oraz metody `dodajPriorytet(int priorytet)` i `usunPriorytet(int indeks)`, które odpowiednio dodają lub usuwają priorytet z listy priorytetów. Dodaj również metodę `najwyzszyPriorytet()` do znajdowania najwyższego priorytetu.
15. Stwórz klasę `Magazyn` z prywatnym polem `iloscProduktow` (jako tablica typu `int`). Dodaj konstruktor, który przyjmuje rozmiar tablicy jako argument. Dodaj metody dostępne (getter i setter) oraz metodę `dodajProdukty(int indeks, int ilosc)`, która dodaje określoną ilość produktów na podanym indeksie. Dodaj również metodę `sumaProduktow()` do obliczania sumy wszystkich produktów w magazynie.