

```

public class Car {                                     // Klasa Car reprezentuje samochód
                                                    // Pola klasy Car

    private String brand;        // Marka samochodu
    private int year;           // Rok produkcji

    public Car(String brand, int year) {              // Konstruktor klasy Car
        this.brand = brand;                          // Inicjalizacja pola brand
        this.year = year;                            // Inicjalizacja pola year
    }

    public String getBrand() {                        // Metoda zwracająca markę samochodu
        return brand;                                // Zwraca markę samochodu
    }

    public int getYear() {                            // Metoda zwracająca rok produkcji samochodu
        return year;                                 // Zwraca rok produkcji samochodu
    }

    // Metoda wyświetlająca informacje o samochodzie
    public void displayInfo() {
        System.out.println("Brand: " + brand + ", Year: " + year); // Wyświetla markę i rok
        produkcji
    }

    public void updateYear(int newYear) {             // Metoda aktualizująca rok produkcji samochodu
        year = newYear;                              // Aktualizuje rok produkcji
    }

    public static void main(String[] args) {          // Główna metoda programu
        Car myCar = new Car("Toyota", 2020);         // Tworzenie obiektu klasy Car
        myCar.displayInfo();                          // Wyświetlanie informacji o samochodzie
        myCar.updateYear(2022);                      // Aktualizacja roku produkcji
        myCar.displayInfo();                          // Wyświetlanie zaktualizowanych informacji o samochodzie
    }
}

```

Przeczytaj powyższy kod. Zastanów się nad nim czy jest dla Ciebie zrozumiały.  
Jeśli tak wykonaj poniższe zadania analogicznie do przykładu powyżej.

**1. Klasa Book:**

- Napisz klasę `Book`, która będzie miała dwa pola: `title` (tytuł książki) i `author` (autor książki).
- Dodaj konstruktor, który inicjalizuje te pola.
- Dodaj metody `getTitle()`, `getAuthor()`, `displayInfo()`, oraz `updateTitle(String newTitle)`.

**2. Klasa Student:**

- Napisz klasę `Student`, która będzie miała dwa pola: `name` (imię studenta) i `id` (numer identyfikacyjny).
- Dodaj konstruktor, który inicjalizuje te pola.
- Dodaj metody `getName()`, `getId()`, `displayInfo()`, oraz `updateId(int newId)`.

**3. Klasa Movie:**

- Napisz klasę `Movie`, która będzie miała dwa pola: `title` (tytuł filmu) i `year` (rok produkcji).
- Dodaj konstruktor, który inicjalizuje te pola.
- Dodaj metody `getTitle()`, `getYear()`, `displayInfo()`, oraz `updateYear(int newYear)`.

**4. Klasa Person:**

- Napisz klasę `Person`, która będzie miała dwa pola: `firstName` (imię) i `lastName` (nazwisko).
- Dodaj konstruktor, który inicjalizuje te pola.
- Dodaj metody `getFirstName()`, `getLastName()`, `displayInfo()`, oraz `updateLastName(String newLastName)`.

**5. Klasa Product:**

- Napisz klasę `Product`, która będzie miała dwa pola: `name` (nazwa produktu) i `price` (cena).
- Dodaj konstruktor, który inicjalizuje te pola.
- Dodaj metody `getName()`, `getPrice()`, `displayInfo()`, oraz `updatePrice(double newPrice)`.

**6. Klasa Employee:**

- Napisz klasę `Employee`, która będzie miała dwa pola: `name` (imię pracownika) i `salary` (wynagrodzenie).
- Dodaj konstruktor, który inicjalizuje te pola.
- Dodaj metody `getName()`, `getSalary()`, `displayInfo()`, oraz `updateSalary(double newSalary)`.

**7. Klasa City:**

- Napisz klasę `City`, która będzie miała dwa pola: `name` (nazwa miasta) i `population` (populacja).
- Dodaj konstruktor, który inicjalizuje te pola.
- Dodaj metody `getName()`, `getPopulation()`, `displayInfo()`, oraz `updatePopulation(int newPopulation)`.

**8. Klasa Course:**

- Napisz klasę `Course`, która będzie miała dwa pola: `courseName` (nazwa kursu) i `credits` (liczba punktów ECTS).
- Dodaj konstruktor, który inicjalizuje te pola.
- Dodaj metody `getCourseName()`, `getCredits()`, `displayInfo()`, oraz `updateCredits(int newCredits)`.

#### 9. Klasa `Animal`:

- Napisz klasę `Animal`, która będzie miała dwa pola: `species` (gatunek) i `age` (wiek).
- Dodaj konstruktor, który inicjalizuje te pola.
- Dodaj metody `getSpecies()`, `getAge()`, `displayInfo()`, oraz `updateAge(int newAge)`.

#### 10. Klasa `Laptop`:

- Napisz klasę `Laptop`, która będzie miała dwa pola: `brand` (marka) i `model` (model).
- Dodaj konstruktor, który inicjalizuje te pola.
- Dodaj metody `getBrand()`, `getModel()`, `displayInfo()`, oraz `updateModel(String newModel)`.

Z kodu projektów IntelliJ wyodrębnij jedynie swoje pliki z rozszerzeniem `.java` z katalogu `/src` Spakuj je i wyślij na Moodle – to rozwiązanie Twoich zadań.

Zadanie domowe.

W zeszycie własnoręcznie wpisz kod będący realizacją na poniższych zadań:

Oczywiście! Oto kolejne 10 przykładów zadań polegających na tworzeniu analogicznych klas i ich użyciu:

#### 1. Klasa `House`:

- Napisz klasę `House`, która będzie miała dwa pola: `address` (adres) i `size` (wielkość w metrach kwadratowych).
- Dodaj konstruktor, który inicjalizuje te pola.
- Dodaj metody `getAddress()`, `getSize()`, `displayInfo()`, oraz `updateSize(int newSize)`.

#### 2. Klasa `Bike`:

- Napisz klasę `Bike`, która będzie miała dwa pola: `brand` (marka) i `type` (typ roweru, np. górski, szosowy).
- Dodaj konstruktor, który inicjalizuje te pola.
- Dodaj metody `getBrand()`, `getType()`, `displayInfo()`, oraz `updateType(String newType)`.

#### 3. Klasa `Game`:

- Napisz klasę `Game`, która będzie miała dwa pola: `title` (tytuł gry) i `genre` (gatunek gry).
- Dodaj konstruktor, który inicjalizuje te pola.
- Dodaj metody `getTitle()`, `getGenre()`, `displayInfo()`, oraz `updateGenre(String newGenre)`.

#### 4. Klasa `Restaurant`:

- Napisz klasę `Restaurant`, która będzie miała dwa pola: `name` (nazwa restauracji) i `rating` (ocena w skali 1-5).
- Dodaj konstruktor, który inicjalizuje te pola.
- Dodaj metody `getName()`, `getRating()`, `displayInfo()`, oraz `updateRating(int newRating)`.

#### 5. Klasa Phone:

- Napisz klasę `Phone`, która będzie miała dwa pola: `brand` (marka) i `model` (model telefonu).
- Dodaj konstruktor, który inicjalizuje te pola.
- Dodaj metody `getBrand()`, `getModel()`, `displayInfo()`, oraz `updateModel(String newModel)`.

#### 6. Klasa Teacher:

- Napisz klasę `Teacher`, która będzie miała dwa pola: `name` (imię nauczyciela) i `subject` (przedmiot, którego uczy).
- Dodaj konstruktor, który inicjalizuje te pola.
- Dodaj metody `getName()`, `getSubject()`, `displayInfo()`, oraz `updateSubject(String newSubject)`.

#### 7. Klasa Library:

- Napisz klasę `Library`, która będzie miała dwa pola: `name` (nazwa biblioteki) i `location` (lokalizacja).
- Dodaj konstruktor, który inicjalizuje te pola.
- Dodaj metody `getName()`, `getLocation()`, `displayInfo()`, oraz `updateLocation(String newLocation)`.

#### 8. Klasa Hotel:

- Napisz klasę `Hotel`, która będzie miała dwa pola: `name` (nazwa hotelu) i `stars` (liczba gwiazdek).
- Dodaj konstruktor, który inicjalizuje te pola.
- Dodaj metody `getName()`, `getStars()`, `displayInfo()`, oraz `updateStars(int newStars)`.

#### 9. Klasa Song:

- Napisz klasę `Song`, która będzie miała dwa pola: `title` (tytuł piosenki) i `artist` (wykonawca).
- Dodaj konstruktor, który inicjalizuje te pola.
- Dodaj metody `getTitle()`, `getArtist()`, `displayInfo()`, oraz `updateArtist(String newArtist)`.

#### 10. Klasa Magazine:

- Napisz klasę `Magazine`, która będzie miała dwa pola: `title` (tytuł magazynu) i `issueNumber` (numer wydania).
- Dodaj konstruktor, który inicjalizuje te pola.
- Dodaj metody `getTitle()`, `getIssueNumber()`, `displayInfo()`, oraz `updateIssueNumber(int newIssueNumber)`.