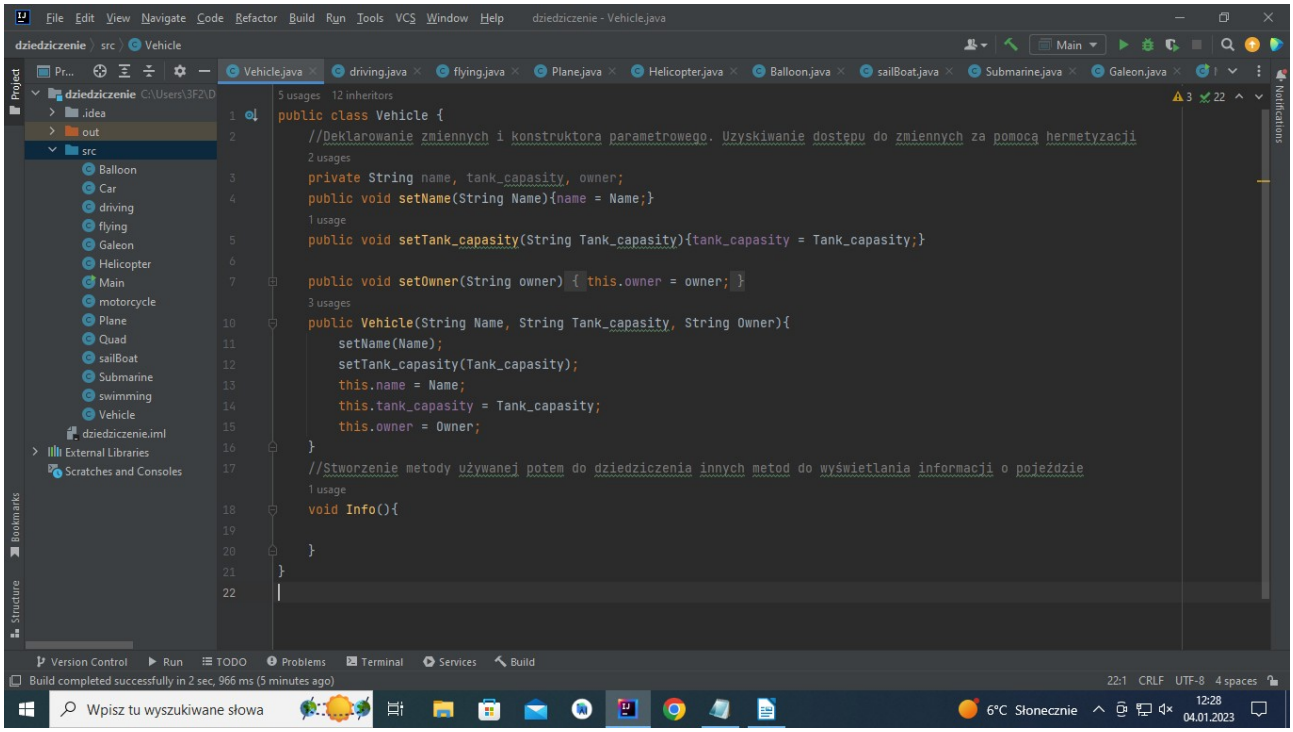
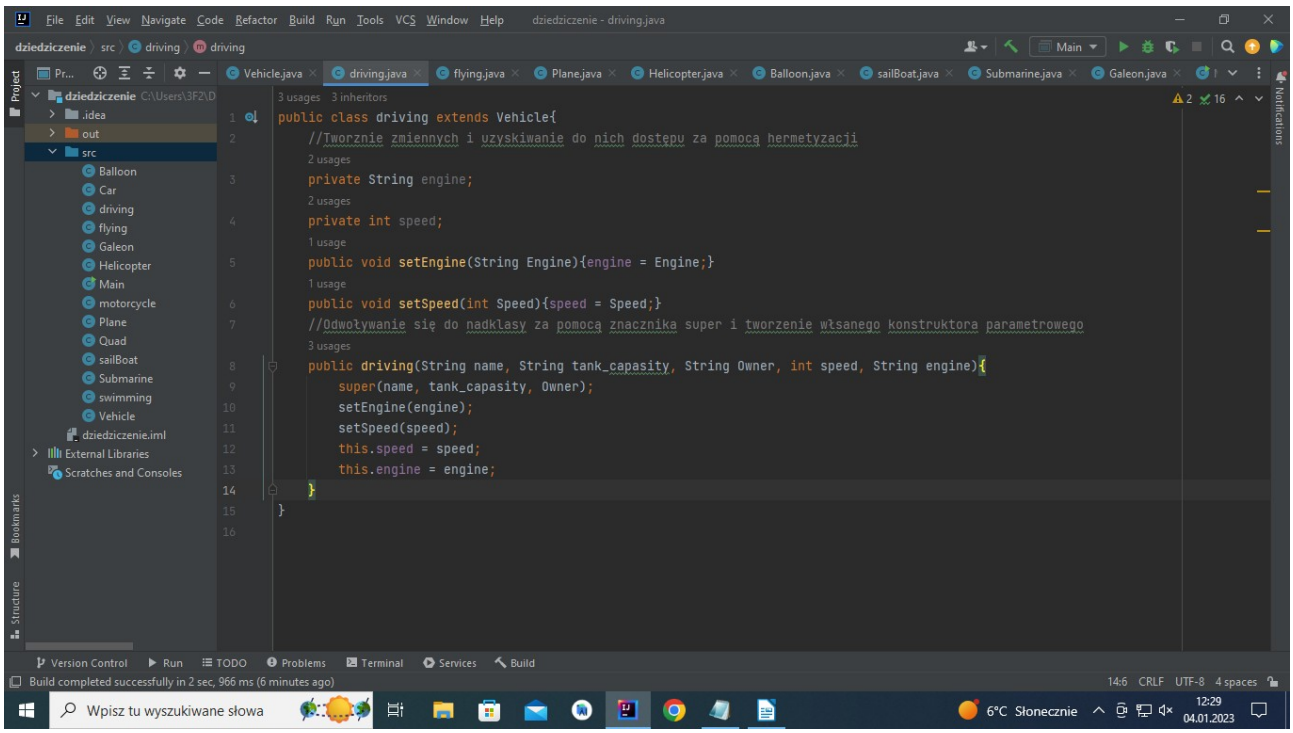
Michał Wiliński
Dokumentacja projektu z obiektowości

Spis Treści:

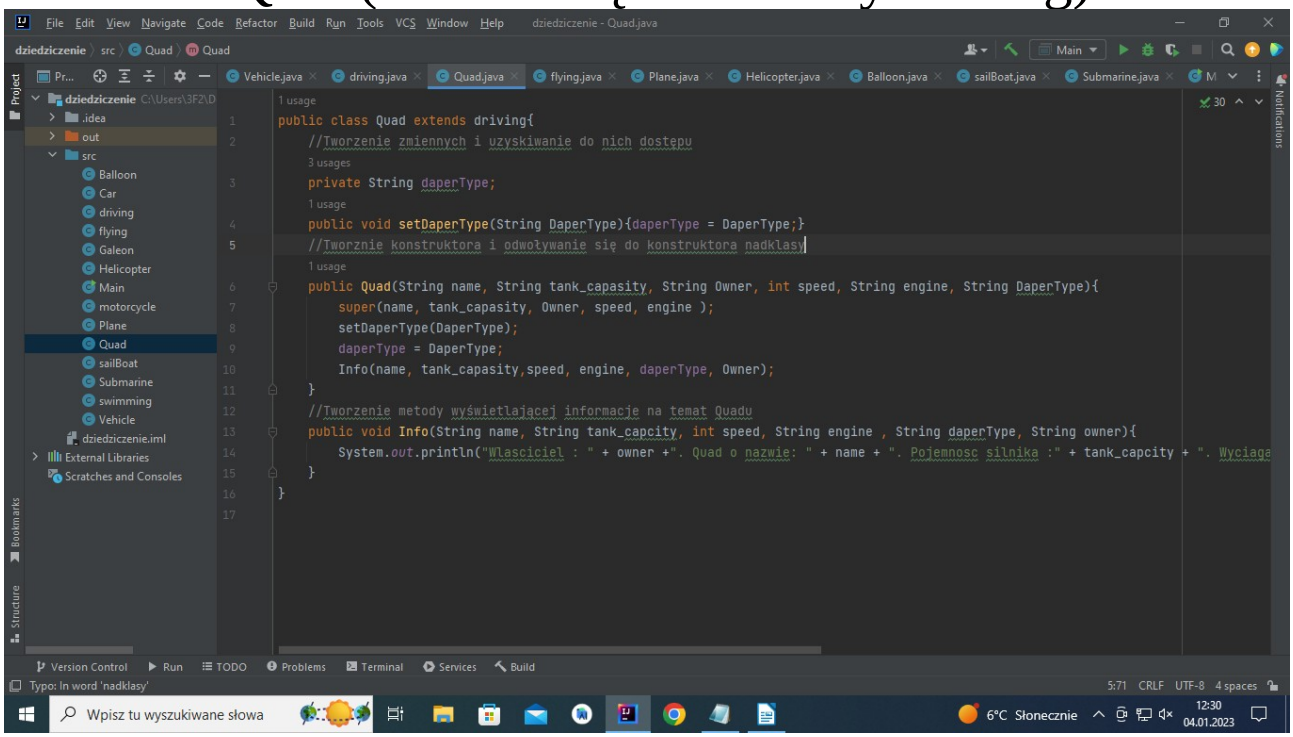
Nadklasa Vehicle:



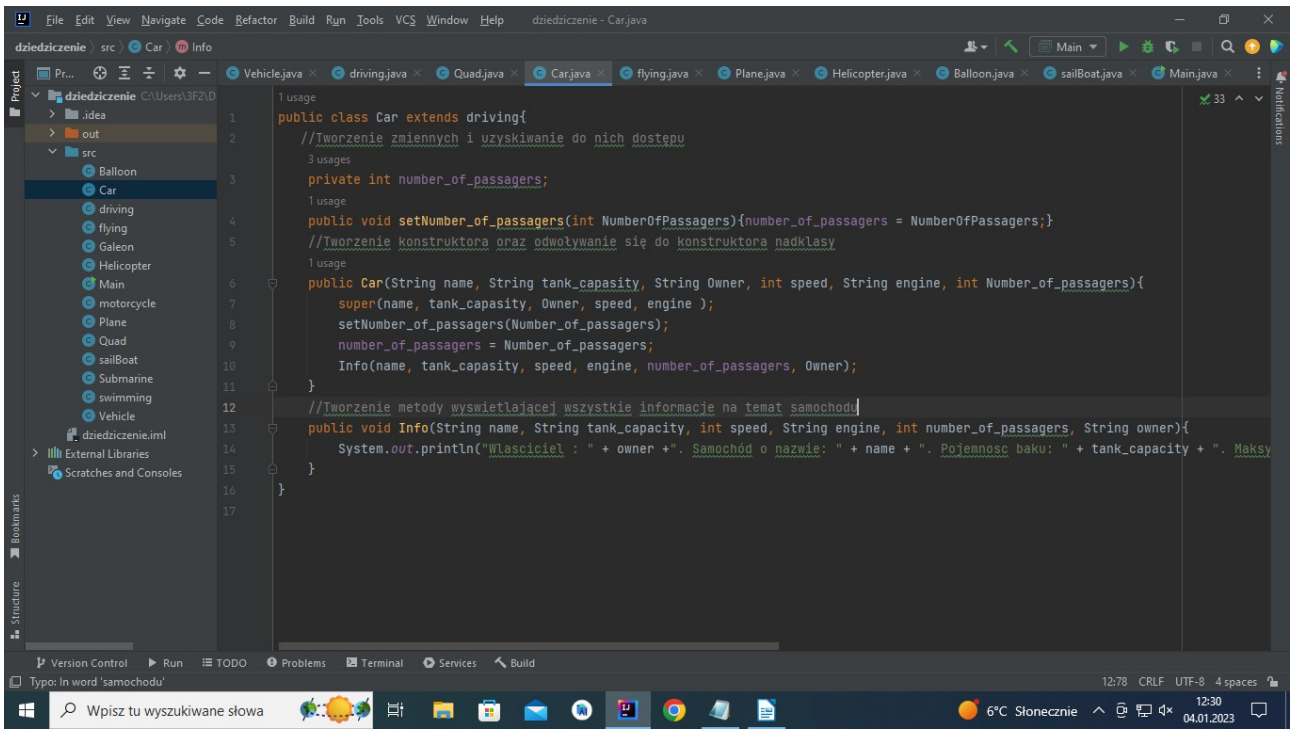
Podklasa driving(dziedzicząca od Vehicle)



Podklasa Quad(dziedzicząca od klasy driving)

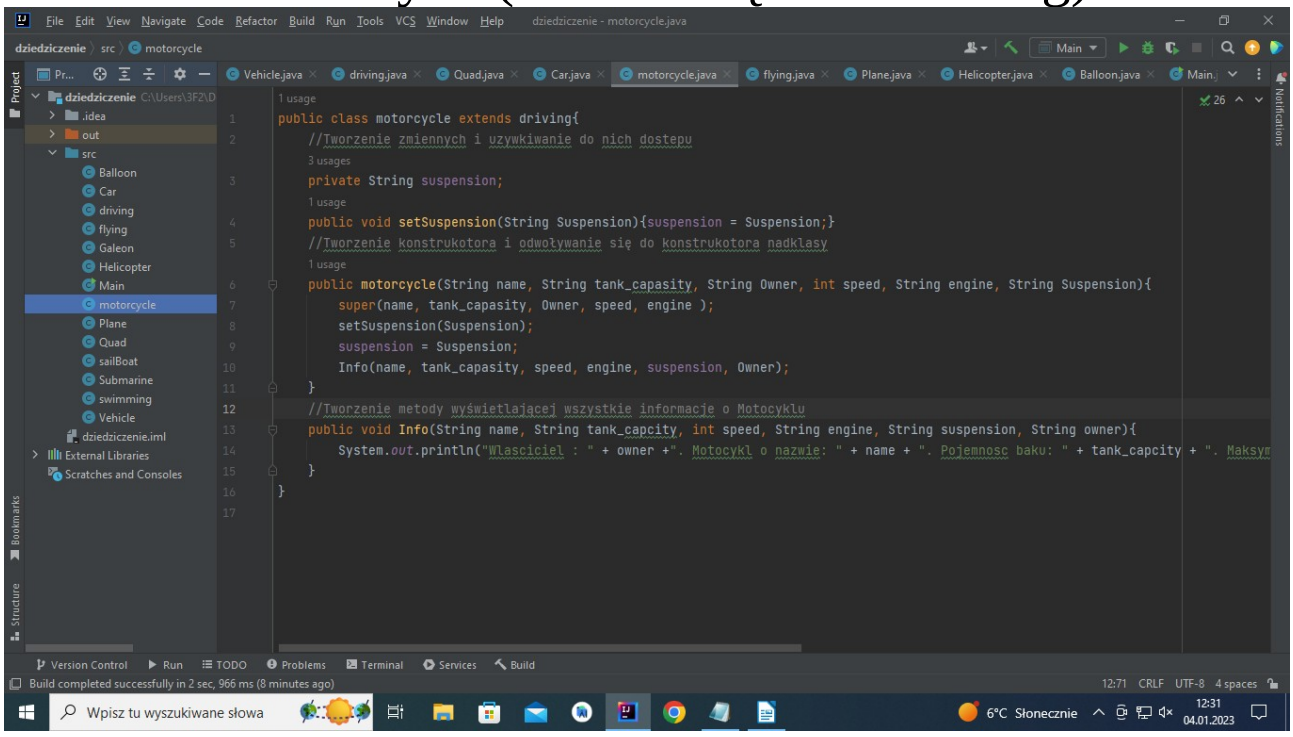


Podklasa Car(dziedzicząca od Vehicle)



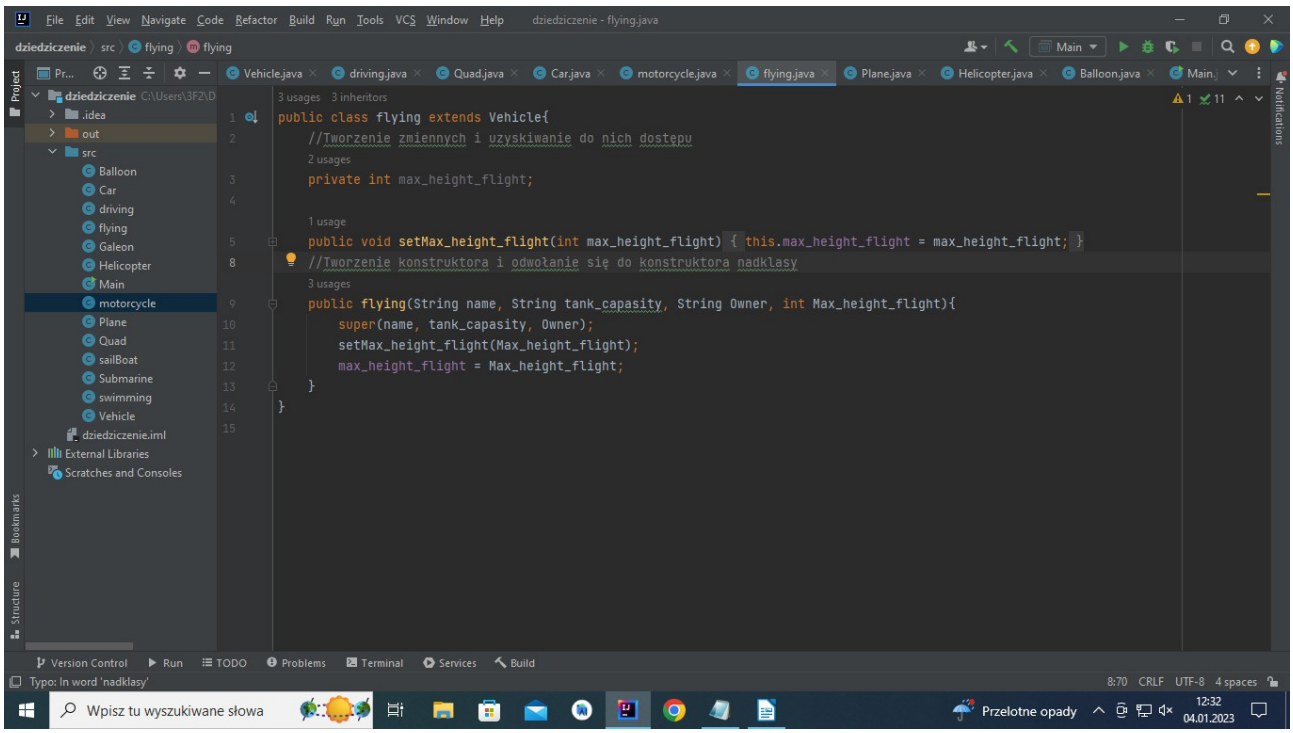
```java
public class Car extends driving{
    //Tworzenie zmiennych i uzyskiwanie do nich dostepu
    private int number_of_passagers;
    public void setNumber_of_passagers(int NumberOfPassagers){number_of_passagers = NumberOfPassagers;}
    //Tworzenie konstruktora oraz odwoływanie się do konstruktora nadklasy
    public Car(String name, String tank_capasity, String Owner, int speed, String engine, int Number_of_passagers){
        super(name, tank_capasity, Owner, speed, engine );
        setNumber_of_passagers(Number_of_passagers);
        number_of_passagers = Number_of_passagers;
        Info(name, tank_capasity, speed, engine, number_of_passagers, Owner);
    }
    //Tworzenie metody wyswietlającej wszystkie informacje na temat samochodu
    public void Info(String name, String tank_capacity, int speed, String engine, int number_of_passagers, String owner){
        System.out.println("Wlasciciel : " + owner +". Samochód o nazwie: " + name + ". Pojemnosc baku: " + tank_capacity + ". Maksy
    }
}
```

Podklasa motorcycle(dziedzicząca od driving)

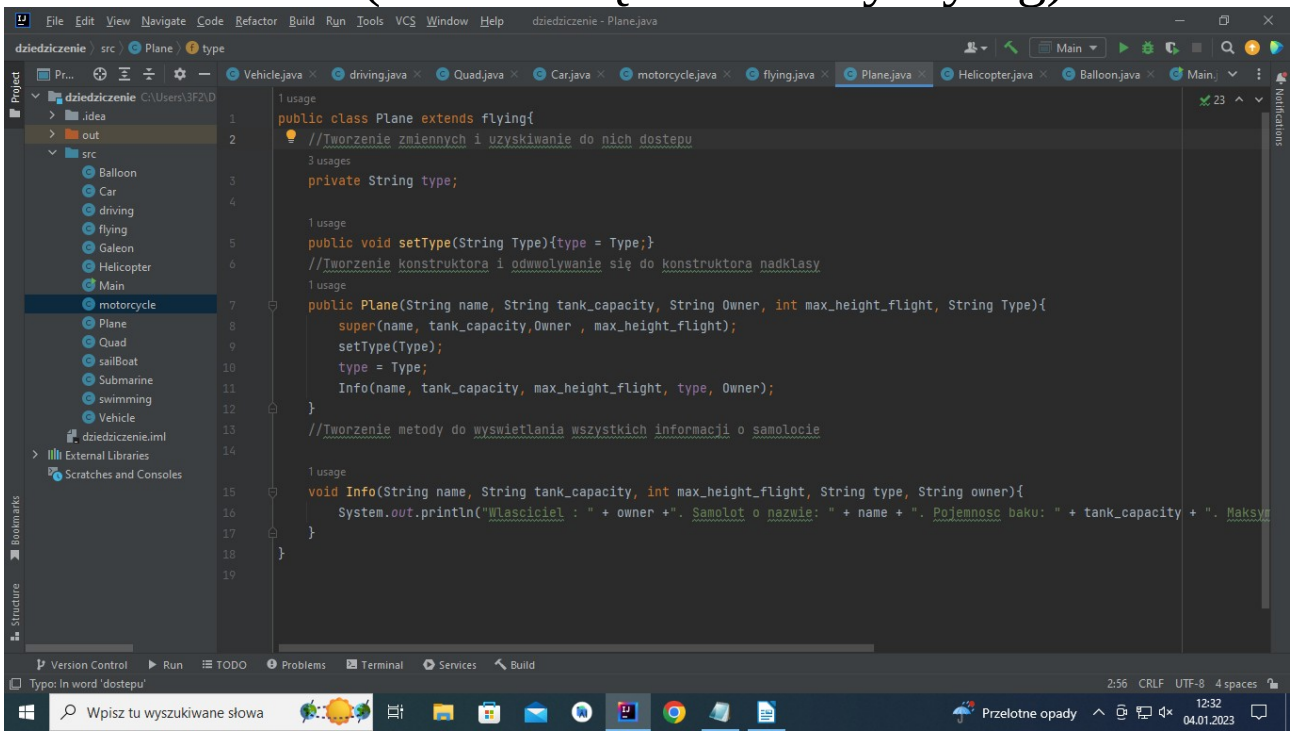


```java
public class motorcycle extends driving{
    //Tworzenie zmiennych i uzywkiwanie do nich dostepu
    private String suspension;
    public void setSuspension(String Suspension){suspension = Suspension;}
    //Tworzenie konstrukotora i odwoływanie się do konstrukotora nadklasy
    public motorcycle(String name, String tank_capasity, String Owner, int speed, String engine, String Suspension){
        super(name, tank_capasity, Owner, speed, engine );
        setSuspension(Suspension);
        suspension = Suspension;
        Info(name, tank_capasity, speed, engine, suspension, Owner);
    }
    //Tworzenie metody wyświetlającej wszystkie informacje o Motocyklu
    public void Info(String name, String tank_capcity, int speed, String engine, String suspension, String owner){
        System.out.println("Wlasciciel : " + owner +". Motocykl o nazwie: " + name + ". Pojemnosc baku: " + tank_capcity + ". Maksym
    }
}
```

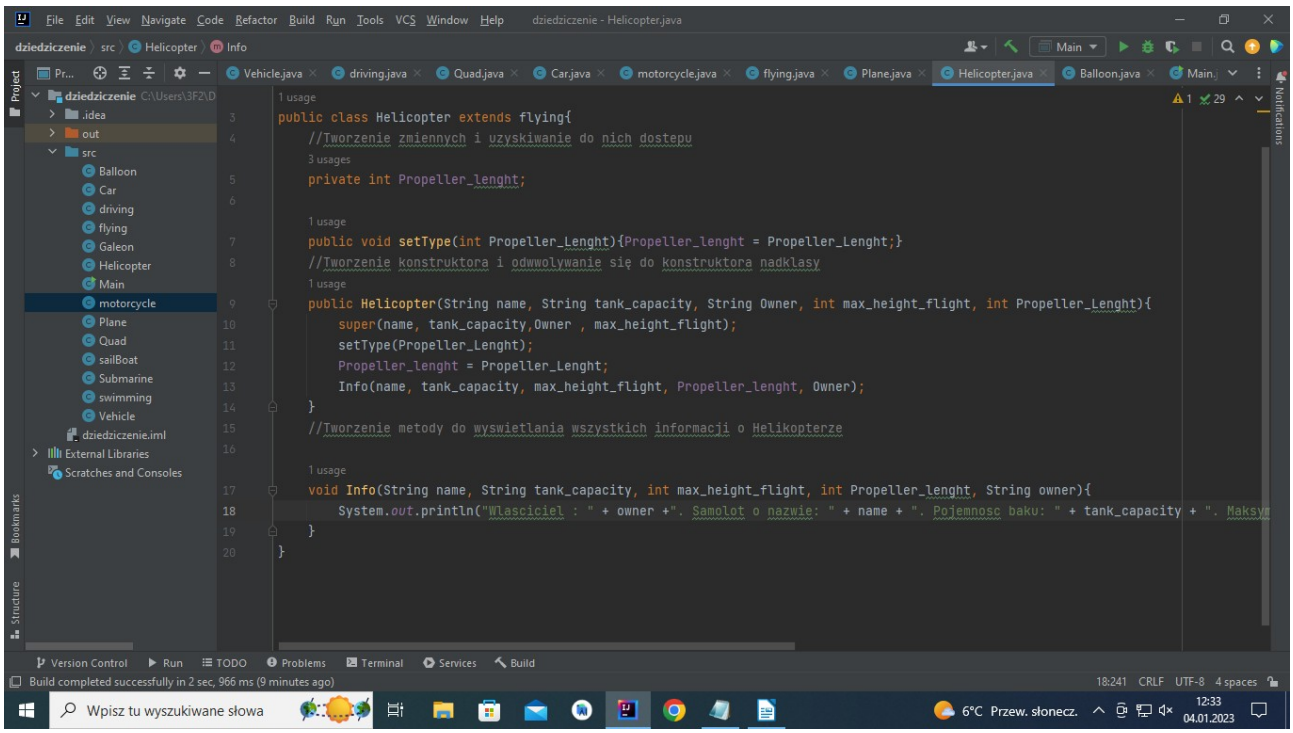
Podklasa Flying(dziedzicząca od Vehicle)



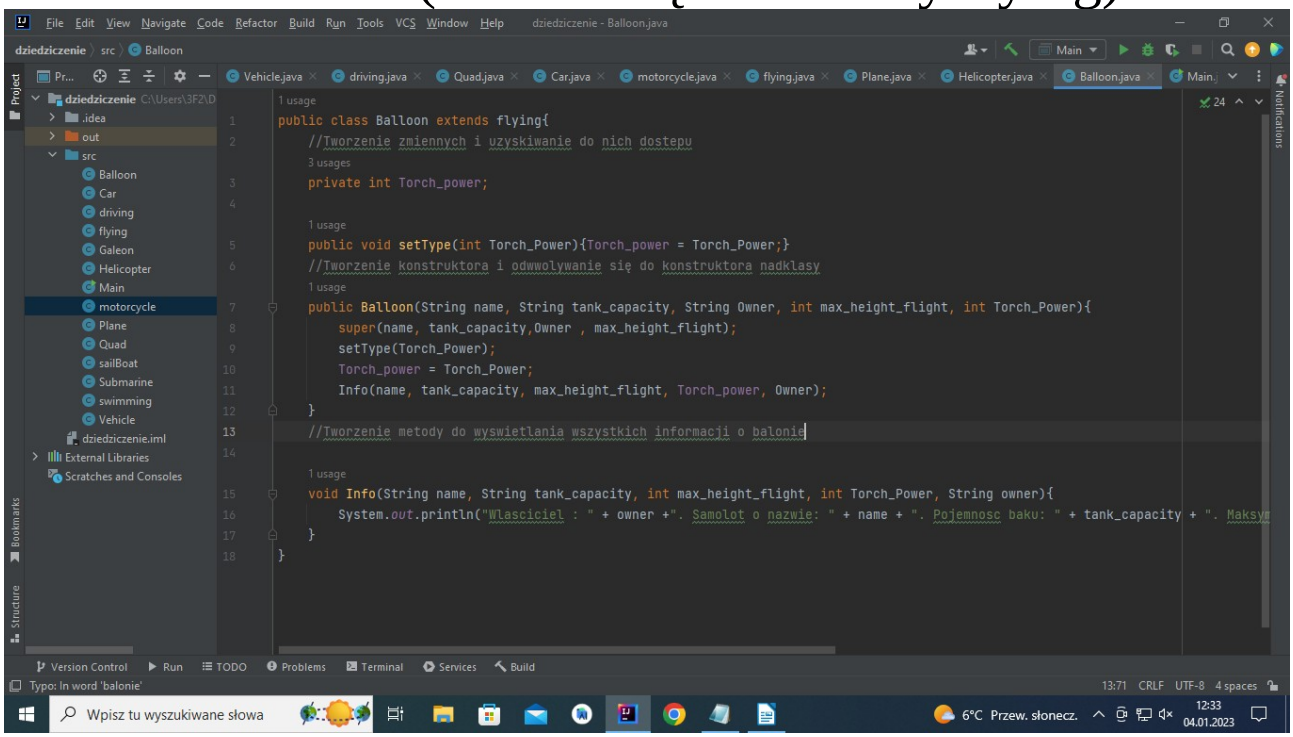
Podklasa Plane(dziedzicząca od klasy flying)



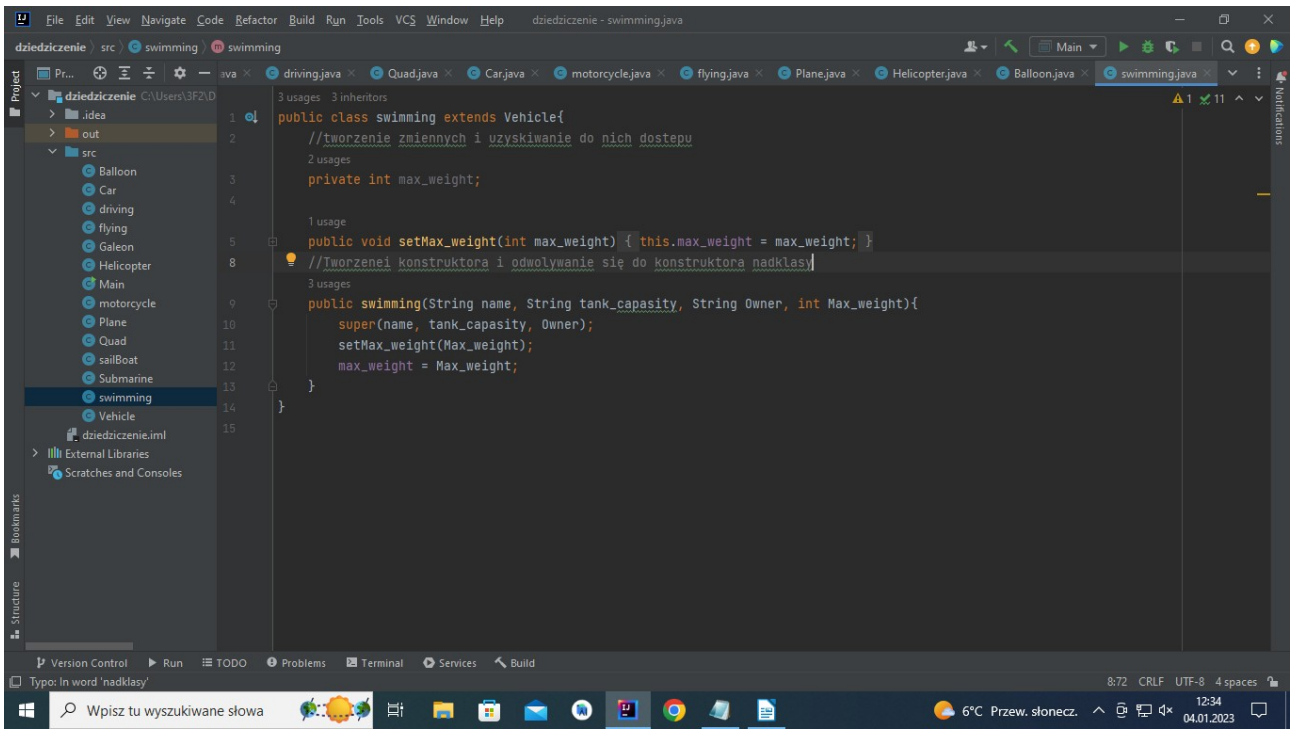
Podklasa Helicopter(dziedzicząca od klasy flying)



Podklasa Balloon(dziedzicząca od klasy flying)



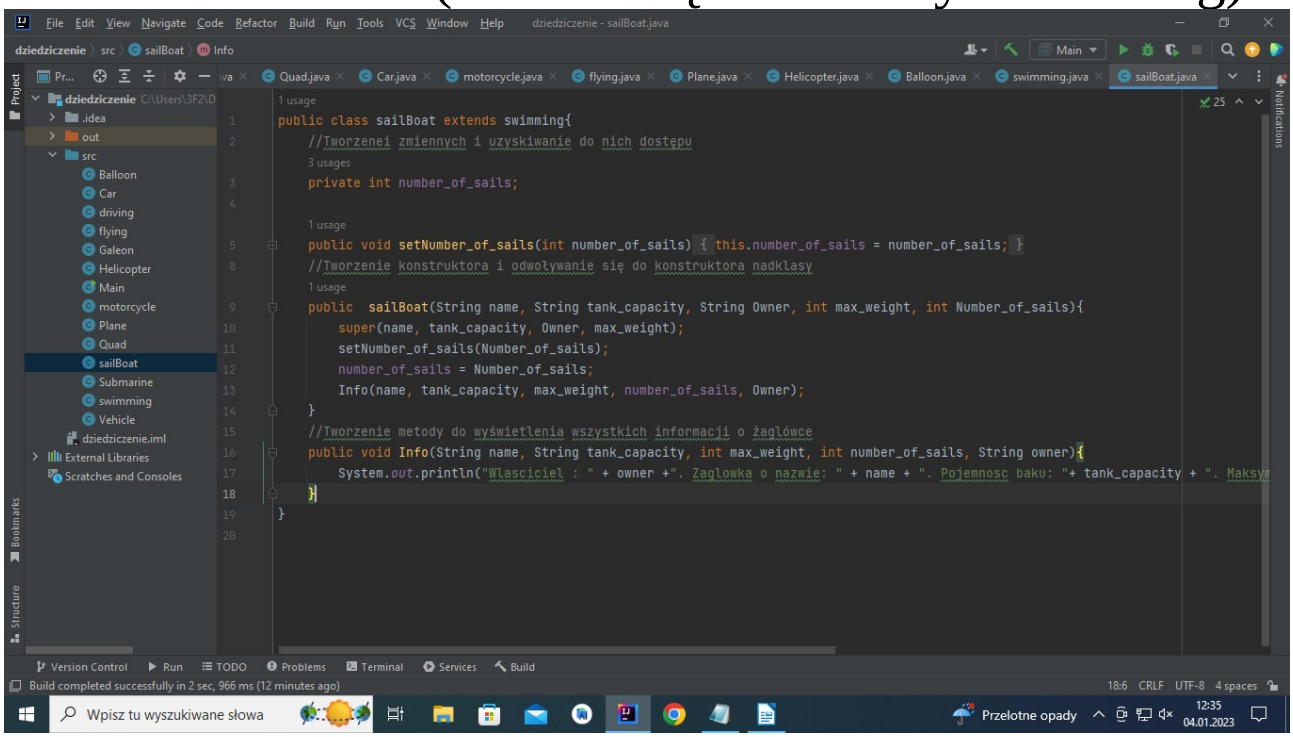
Podklasa Swimming(dziedzicząca od klasy Vehicle)



```java
public class swimming extends Vehicle{
    //tworzenie zmiennych i uzyskiwanie do nich dostepu
    private int max_weight;

    public void setMax_weight(int max_weight) { this.max_weight = max_weight; }
    //Tworzenei konstruktora i odwolywanie się do konstruktora nadklasy
    public swimming(String name, String tank_capasity, String Owner, int Max_weight){
        super(name, tank_capasity, Owner);
        setMax_weight(Max_weight);
        max_weight = Max_weight;
    }
}
```

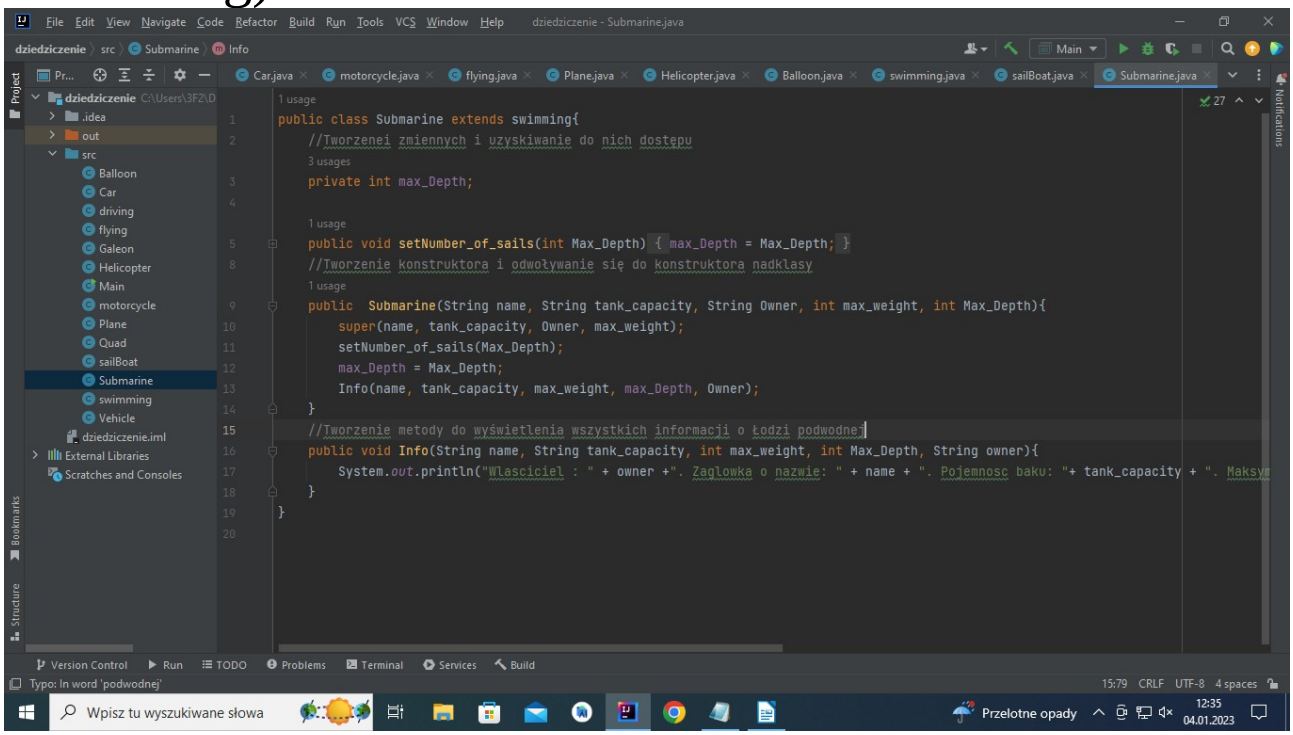
Podklasa SailBoat(dziedzicząca od klasy swimming)



```java
public class sailBoat extends swimming{
    //Tworzenei zmiennych i uzyskiwanie do nich dostepu
    private int number_of_sails;

    public void setNumber_of_sails(int number_of_sails) { this.number_of_sails = number_of_sails; }
    //Tworzenie konstruktora i odwoływanie się do konstruktora nadklasy
    public  sailBoat(String name, String tank_capacity, String Owner, int max_weight, int Number_of_sails){
        super(name, tank_capacity, Owner, max_weight);
        setNumber_of_sails(Number_of_sails);
        number_of_sails = Number_of_sails;
        Info(name, tank_capacity, max_weight, number_of_sails, Owner);
    }
    //Tworzenie metody do wyświetlenia wszystkich informacji o żaglówce
    public void Info(String name, String tank_capacity, int max_weight, int number_of_sails, String owner){
        System.out.println("Wlasciciel : " + owner +". Zaglowka o nazwie: " + name + ". Pojemnosc baku: "+ tank_capacity + ". Maksym
    }
}
```

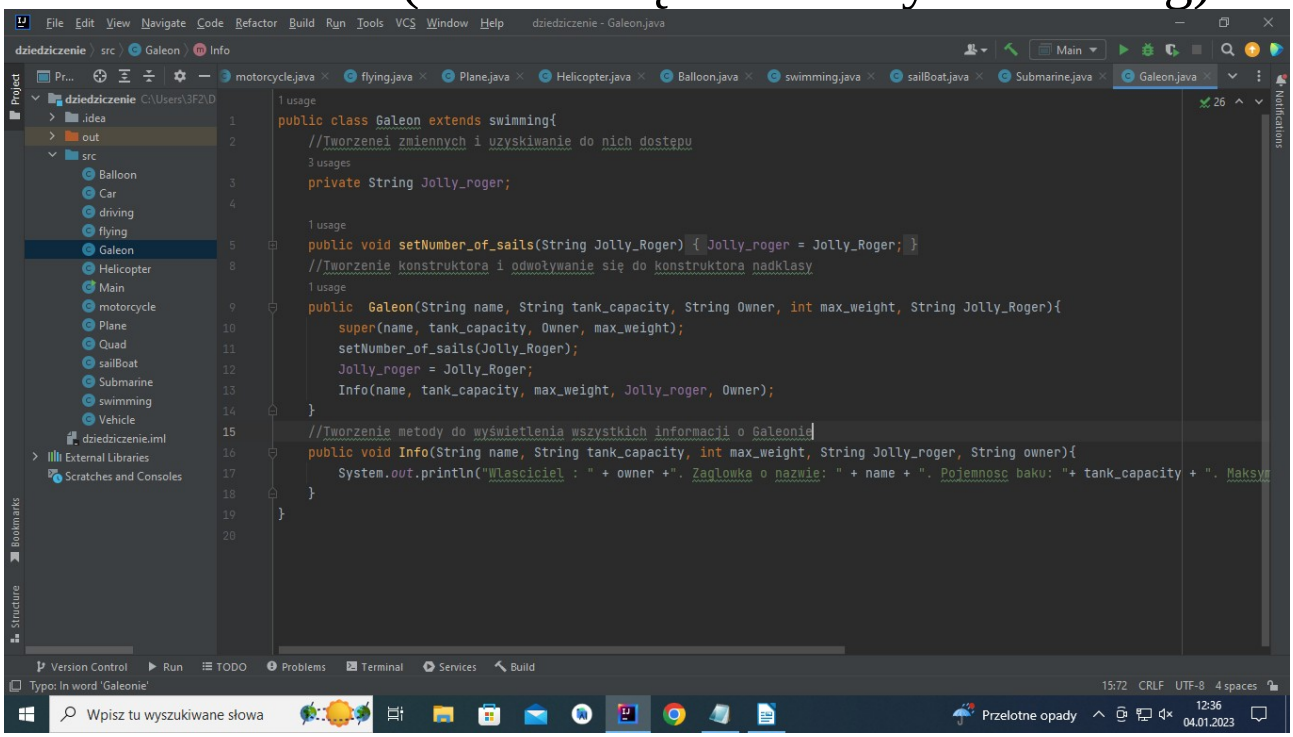
Podklasa Submarine(dziedzicząca od klasy swimming)



```java
public class Submarine extends swimming{
    //Tworzenei zmiennych i uzyskiwanie do nich dostępu
    private int max_Depth;

    public void setNumber_of_sails(int Max_Depth) { max_Depth = Max_Depth; }
    //Tworzenie konstruktora i odwoływanie się do konstruktora nadklasy
    public  Submarine(String name, String tank_capacity, String Owner, int max_weight, int Max_Depth){
        super(name, tank_capacity, Owner, max_weight);
        setNumber_of_sails(Max_Depth);
        max_Depth = Max_Depth;
        Info(name, tank_capacity, max_weight, max_Depth, Owner);
    }
    //Tworzenie metody do wyświetlenia wszystkich informacji o Łodzi podwodnej
    public void Info(String name, String tank_capacity, int max_weight, int Max_Depth, String owner){
        System.out.println("Wlasciciel : " + owner +". Zaglowka o nazwie: " + name + ". Pojemnosc baku: "+ tank_capacity + ". Maksym
    }
}
```

Podklasa Galeon(dziedzicząca od klasy swimming)



```java
public class Galeon extends swimming{
    //Tworzenei zmiennych i uzyskiwanie do nich dostępu
    private String Jolly_roger;

    public void setNumber_of_sails(String Jolly_Roger) { Jolly_roger = Jolly_Roger; }
    //Tworzenie konstruktora i odwoływanie się do konstruktora nadklasy
    public  Galeon(String name, String tank_capacity, String Owner, int max_weight, String Jolly_Roger){
        super(name, tank_capacity, Owner, max_weight);
        setNumber_of_sails(Jolly_Roger);
        Jolly_roger = Jolly_Roger;
        Info(name, tank_capacity, max_weight, Jolly_roger, Owner);
    }
    //Tworzenie metody do wyświetlenia wszystkich informacji o Galeonie
    public void Info(String name, String tank_capacity, int max_weight, String Jolly_roger, String owner){
        System.out.println("Wlasciciel : " + owner +". Zaglowka o nazwie: " + name + ". Pojemnosc baku: "+ tank_capacity + ". Maksym
    }
}
```

Klasa Main oraz wynik kompilacji