

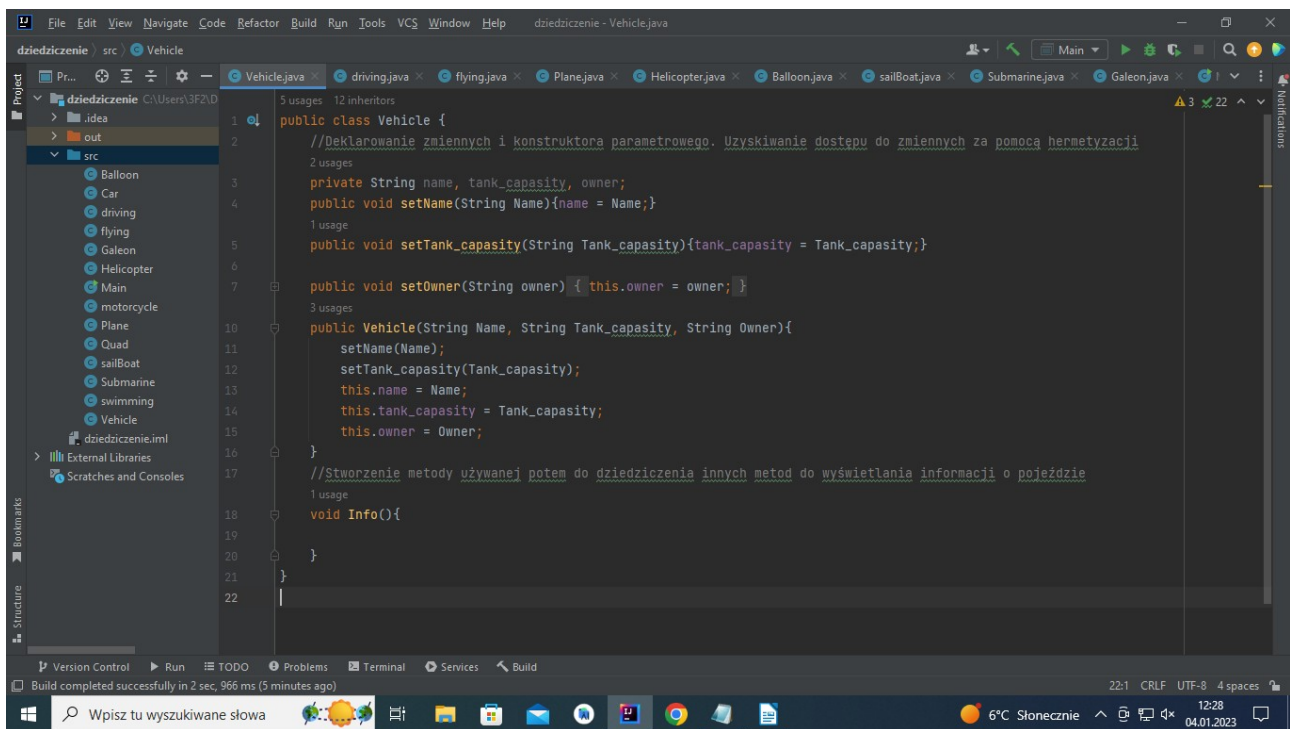
Michał Wiliński

Dokumentacja projektu z obiektowości

Spis Treści:

- Nadklasa Vehicle str. 2
- Podklasa driving i jej podklasy str. 2,5 - 4
- Podklasa flying i jej podklasy str. 4,5 – 6
- Podklasa swimming i jej podklasy 6,5 – 9
- Klasa Main i wyniki kompilacji str. 9,5

Nadklasa Vehicle:



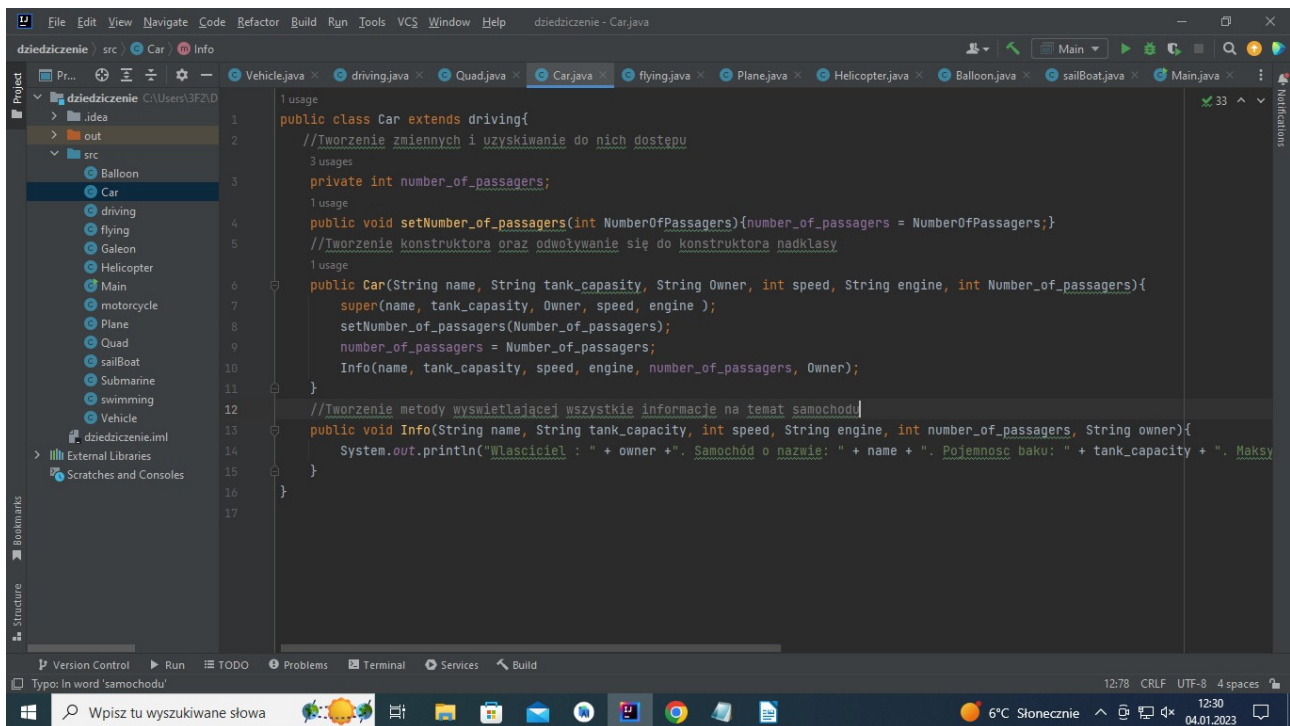
Podklasa driving(dziedzicząca od Vehicle)

```
1 public class driving extends Vehicle{
2     //Tworzenie zmiennych i uzyskiwanie do nich dostępu za pomocą hermetyzacji
3     private String engine;
4     private int speed;
5     public void setEngine(String Engine){engine = Engine;}
6     public void setSpeed(int Speed){speed = Speed;}
7     //Odwołanie się do nadklasy za pomocą znacznika super i tworzenie wisanego konstruktora parametrowego
8     public driving(String name, String tank_capacity, String Owner, int speed, String engine){
9         super(name, tank_capacity, Owner);
10        setEngine(engine);
11        setSpeed(speed);
12        this.speed = speed;
13        this.engine = engine;
14    }
15 }
16
```

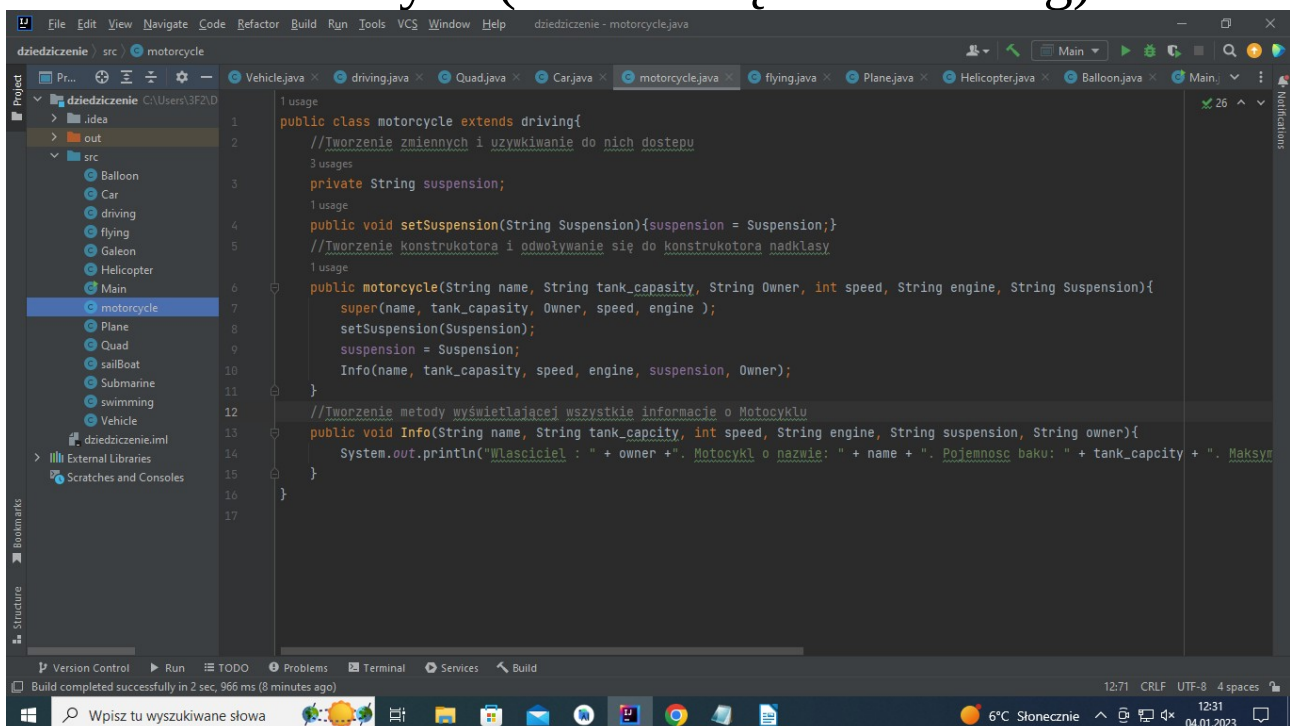
Podklasa Quad(dziedzicząca od klasy driving)

```
1 public class Quad extends driving{
2     //Tworzenie zmiennych i uzyskiwanie do nich dostępu
3     private String daperType;
4     public void setDaperType(String DaperType){daperType = DaperType;}
5     //Tworzenie konstruktora i odwołanie się do konstruktora nadklasy
6     public Quad(String name, String tank_capacity, String Owner, int speed, String engine, String DaperType){
7         super(name, tank_capacity, Owner, speed, engine );
8         setDaperType(DaperType);
9         daperType = DaperType;
10        Info(name, tank_capacity, speed, engine, daperType, Owner);
11    }
12    //Tworzenie metody wyświetlającej informacje na temat Quadu
13    public void Info(String name, String tank_capacity, int speed, String engine, String daperType, String owner){
14        System.out.println("Własciciel : " + owner + ". Quad o nazwie: " + name + ". Pojemność silnika : " + tank_capacity + ". Wyciąg
15    }
16 }
17
```

Podklasa Car(dziedzicząca od Vehicle)



Podklasa motorcycle(dziedzicząca od driving)



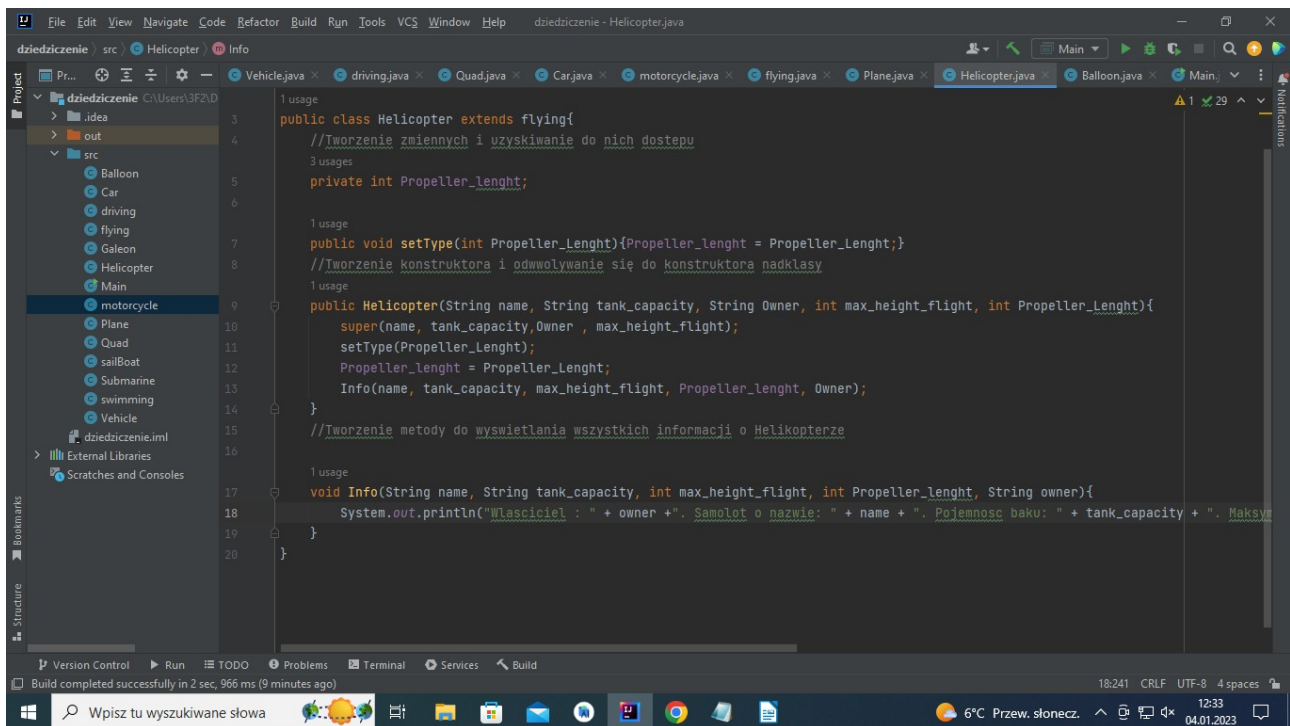
Podklasa Flying(dziedzicząca od Vehicle)

```
1 public class flying extends Vehicle{
2     //Tworzenie zmiennych i uzyskiwanie do nich dostępu
3     private int max_height_flight;
4
5     1 usage
6     public void setMax_height_flight(int max_height_flight){ this.max_height_flight = max_height_flight;}
7     //Tworzenie konstruktora i odwołanie się do konstruktora nadklasy
8     3 usages
9     public flying(String name, String tank_capacity, String Owner, int Max_height_flight){
10         super(name, tank_capacity, Owner);
11         setMax_height_flight(Max_height_flight);
12         max_height_flight = Max_height_flight;
13     }
14 }
15 }
```

Podklasa Plane(dziedzicząca od klasy flying)

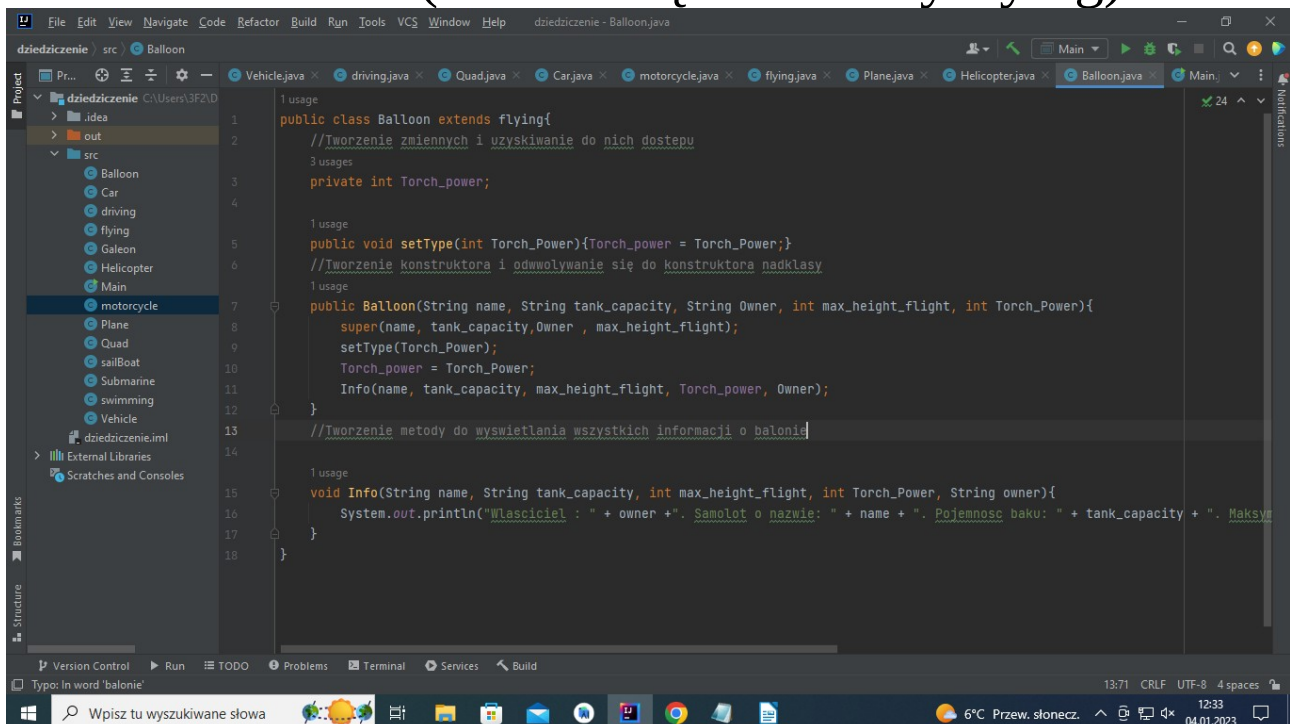
```
1 public class Plane extends flying{
2     //Tworzenie zmiennych i uzyskiwanie do nich dostępu
3     private String type;
4
5     1 usage
6     public void setType(String Type){type = Type;}
7     //Tworzenie konstruktora i odwołanie się do konstruktora nadklasy
8     1 usage
9     public Plane(String name, String tank_capacity, String Owner, int max_height_flight, String Type){
10         super(name, tank_capacity, Owner, max_height_flight);
11         setType(Type);
12         type = Type;
13         Info(name, tank_capacity, max_height_flight, type, Owner);
14     }
15     //Tworzenie metody do wyświetlania wszystkich informacji o samolocie
16     1 usage
17     void Info(String name, String tank_capacity, int max_height_flight, String type, String owner){
18         System.out.println("Własciciel : " + owner + ". Samolot o nazwie: " + name + ". Pojemność baku: " + tank_capacity + ". Maksym.
19     }
20 }
```

Podklasa Helicopter(dziedzicząca od klasy flying)



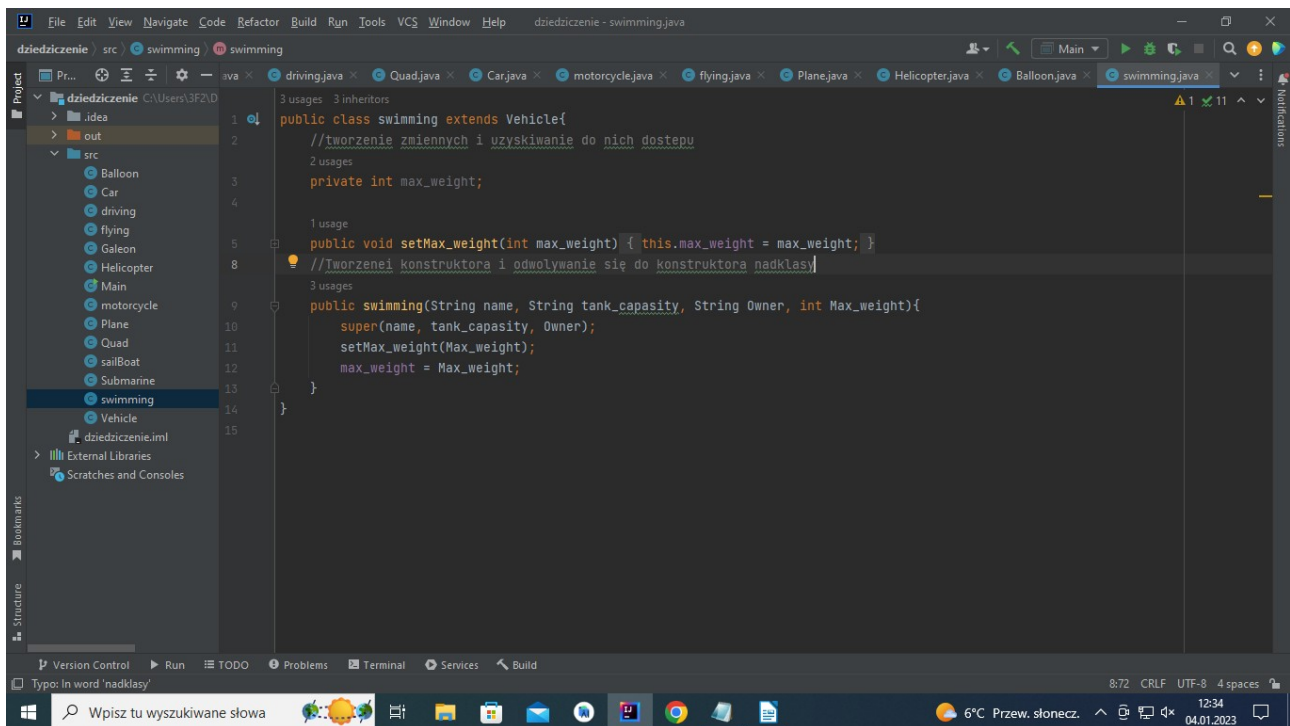
```
1 usage
2 public class Helicopter extends flying{
3     //Tworzenie zmiennych i uzyskiwanie do nich dostępu
4     private int Propeller_Lenght;
5
6     1 usage
7     public void setType(int Propeller_Lenght){Propeller_Lenght = Propeller_Lenght;}
8     //Tworzenie konstruktora i odwoływanie się do konstruktora nadklasy
9     1 usage
10    public Helicopter(String name, String tank_capacity, String Owner, int max_height_flight, int Propeller_Lenght){
11        super(name, tank_capacity, Owner, max_height_flight);
12        setType(Propeller_Lenght);
13        Propeller_Lenght = Propeller_Lenght;
14        Info(name, tank_capacity, max_height_flight, Propeller_Lenght, Owner);
15    }
16    //Tworzenie metody do wyświetlania wszystkich informacji o Helikopterze
17
18    1 usage
19    void Info(String name, String tank_capacity, int max_height_flight, int Propeller_Lenght, String owner){
20        System.out.println("Własciciel : " + owner + ". Samolot o nazwie: " + name + ". Pojemnosc baku: " + tank_capacity + ". Maksymalna wysokość lotu: " + max_height_flight);
21    }
22 }
```

Podklasa Balloon(dziedzicząca od klasy flying)

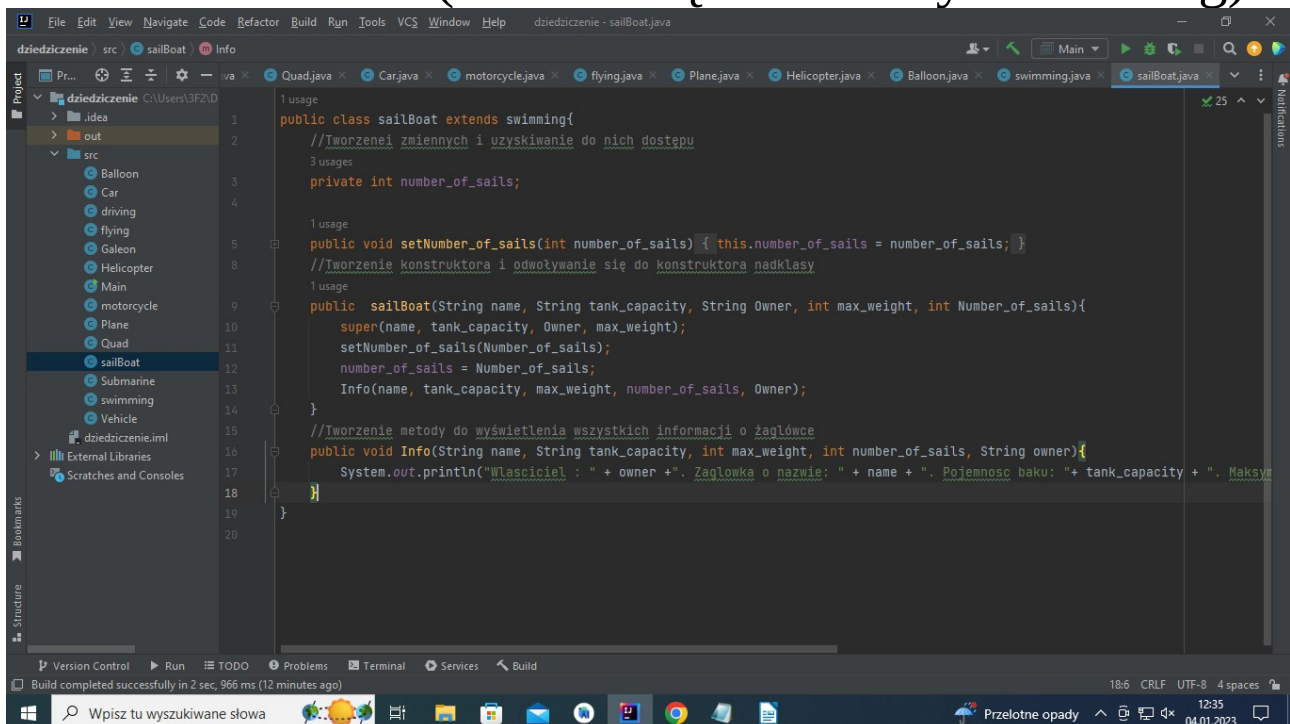


```
1 usage
2 public class Balloon extends flying{
3     //Tworzenie zmiennych i uzyskiwanie do nich dostępu
4     private int Torch_Power;
5
6     1 usage
7     public void setType(int Torch_Power){Torch_Power = Torch_Power;}
8     //Tworzenie konstruktora i odwoływanie się do konstruktora nadklasy
9     1 usage
10    public Balloon(String name, String tank_capacity, String Owner, int max_height_flight, int Torch_Power){
11        super(name, tank_capacity, Owner, max_height_flight);
12        setType(Torch_Power);
13        Torch_Power = Torch_Power;
14        Info(name, tank_capacity, max_height_flight, Torch_Power, Owner);
15    }
16    //Tworzenie metody do wyświetlania wszystkich informacji o balonie
17
18    1 usage
19    void Info(String name, String tank_capacity, int max_height_flight, int Torch_Power, String owner){
20        System.out.println("Własciciel : " + owner + ". Samolot o nazwie: " + name + ". Pojemnosc baku: " + tank_capacity + ". Maksymalna wysokość lotu: " + max_height_flight + ". Moc palenicy: " + Torch_Power);
21    }
22 }
```

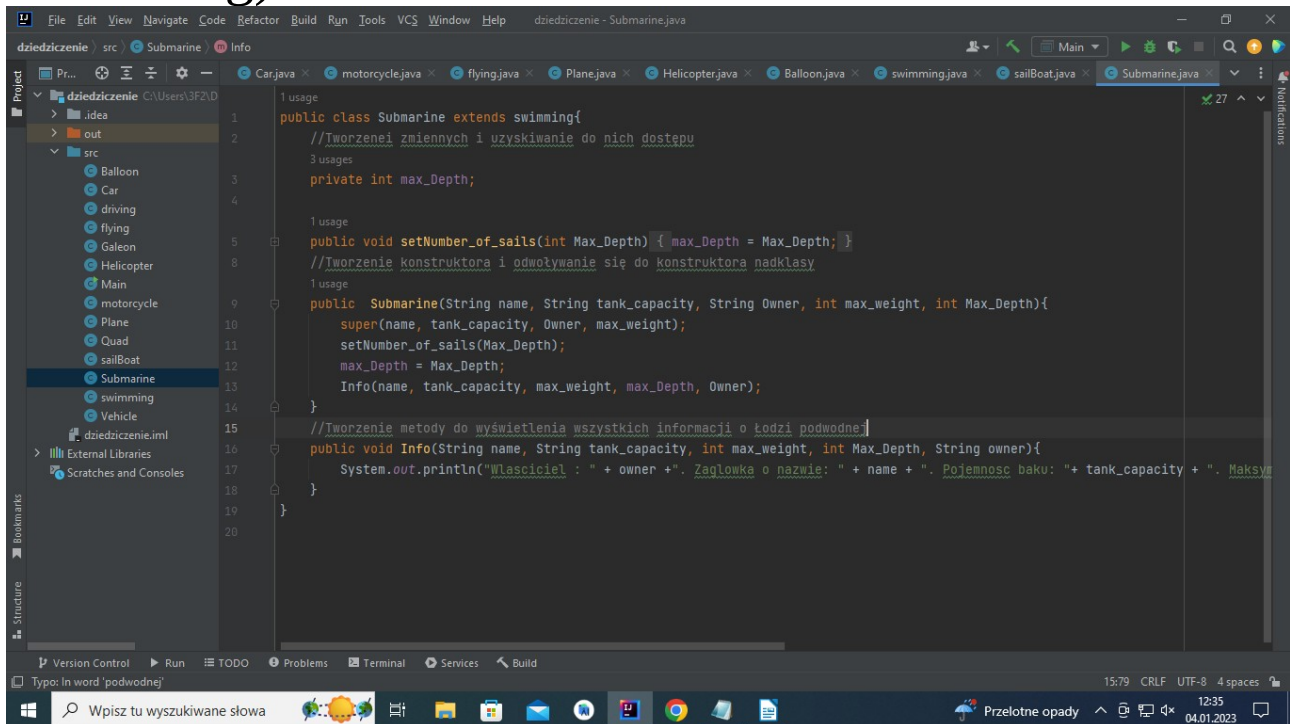
Podklasa Swimming(dziedzicząca od klasy Vehicle)



Podklasa SailBoat(dziedzicząca od klasy swimming)

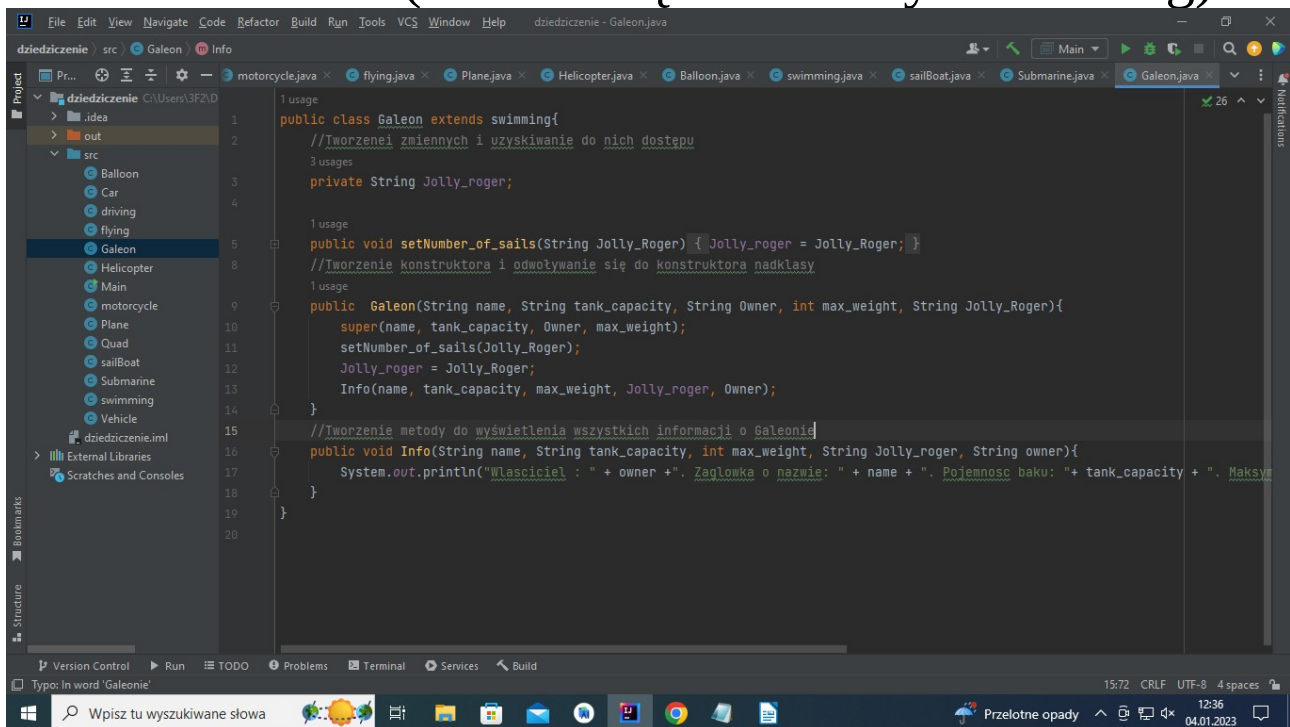


Podklasa Submarine(dziedzicząca od klasy swimming)



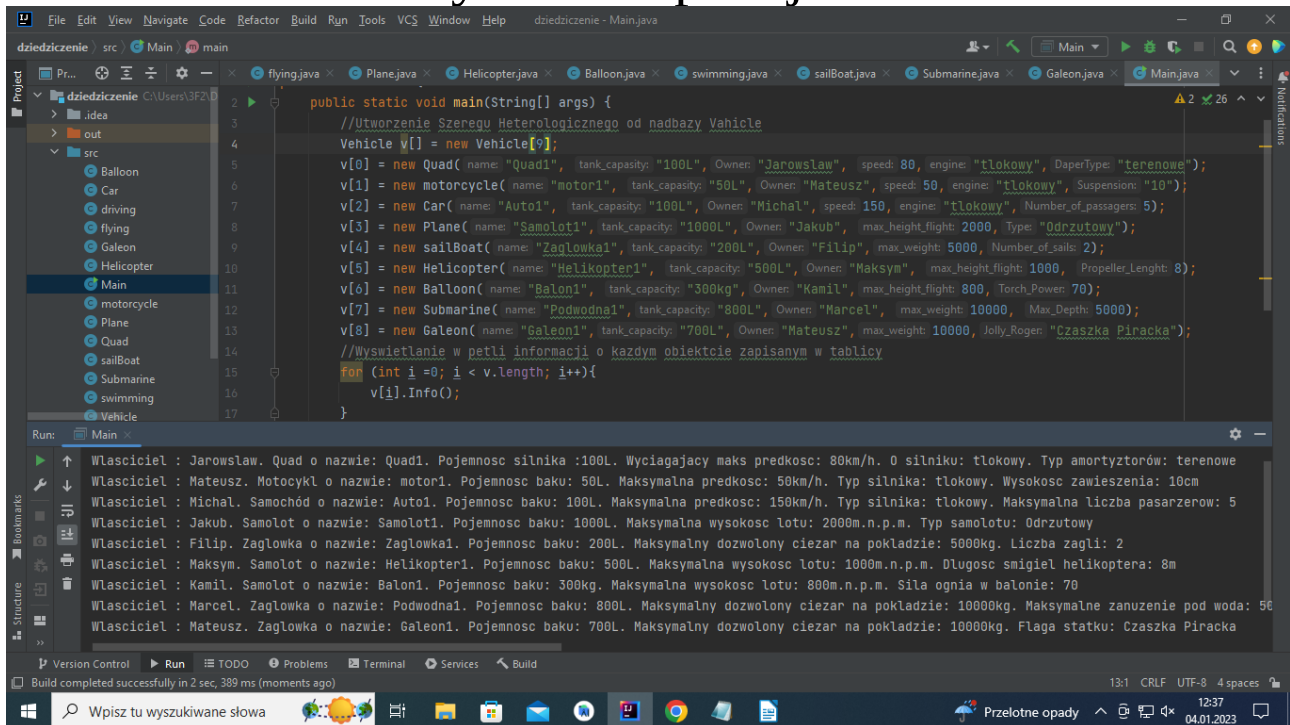
```
1 public class Submarine extends swimming{
2     //Tworzeniei zmiennych i uzyskiwanie do nich dostępu
3     private int max_Depth;
4
5     1 usage
6     public void setNumber_of_sails(int Max_Depth) { max_Depth = Max_Depth; }
7
8     //Tworzenie konstruktora i odwoływanie się do konstruktora nadklasy
9     1 usage
10    public Submarine(String name, String tank_capacity, String Owner, int max_weight, int Max_Depth){
11        super(name, tank_capacity, Owner, max_weight);
12        setNumber_of_sails(Max_Depth);
13        max_Depth = Max_Depth;
14        Info(name, tank_capacity, max_weight, max_Depth, Owner);
15    }
16    //Tworzenie metody do wyświetlenia wszystkich informacji o łodzi podwodnej
17    public void Info(String name, String tank_capacity, int max_weight, int Max_Depth, String owner){
18        System.out.println("Własciciel : " + owner + ". Zagłowa o nazwie: " + name + ". Pojemność baku: " + tank_capacity + ". Maksym
19    }
20 }
```

Podklasa Galeon(dziedzicząca od klasy swimming)



```
1 public class Galeon extends swimming{
2     //Tworzeniei zmiennych i uzyskiwanie do nich dostępu
3     private String Jolly_roger;
4
5     1 usage
6     public void setNumber_of_sails(String Jolly_Roger) { Jolly_roger = Jolly_Roger; }
7
8     //Tworzenie konstruktora i odwoływanie się do konstruktora nadklasy
9     1 usage
10    public Galeon(String name, String tank_capacity, String Owner, int max_weight, String Jolly_Roger){
11        super(name, tank_capacity, Owner, max_weight);
12        setNumber_of_sails(Jolly_Roger);
13        Jolly_roger = Jolly_Roger;
14        Info(name, tank_capacity, max_weight, Jolly_roger, Owner);
15    }
16    //Tworzenie metody do wyświetlenia wszystkich informacji o Galeonie
17    public void Info(String name, String tank_capacity, int max_weight, String Jolly_roger, String owner){
18        System.out.println("Własciciel : " + owner + ". Zagłowa o nazwie: " + name + ". Pojemność baku: " + tank_capacity + ". Maksym
19    }
20 }
```


Klasa Main oraz wynik kompilacji



The screenshot shows an IDE window titled "dziedziczenie - Main.java". The editor displays the following Java code:

```
public static void main(String[] args) {  
    //Utworzenie Szeregu Heterologicznego od nadbazy Vehicle  
    Vehicle v[] = new Vehicle[9];  
    v[0] = new Quad( name: "Quad1", tank_capacity: "100L", Owner: "Jaroslaw", speed: 80, engine: "tlokowy", DaperType: "terenowe");  
    v[1] = new motorcycle( name: "motor1", tank_capacity: "50L", Owner: "Mateusz", speed: 50, engine: "tlokowy", Suspension: "10");  
    v[2] = new Car( name: "Auto1", tank_capacity: "100L", Owner: "Michal", speed: 150, engine: "tlokowy", Number_of_passagers: 5);  
    v[3] = new Plane( name: "Samolot1", tank_capacity: "1000L", Owner: "Jakub", max_height_flight: 2000, Type: "Odrzutowy");  
    v[4] = new sailBoat( name: "Zaglowka1", tank_capacity: "200L", Owner: "Filip", max_weight: 5000, Number_of_sails: 2);  
    v[5] = new Helicopter( name: "Helikopten1", tank_capacity: "500L", Owner: "Maksym", max_height_flight: 1000, Propeller_Length: 8);  
    v[6] = new Balloon( name: "Balon1", tank_capacity: "300kg", Owner: "Kamil", max_height_flight: 800, Torch_Power: 70);  
    v[7] = new Submarine( name: "Podwodna1", tank_capacity: "800L", Owner: "Marcel", max_weight: 10000, Max_Depth: 5000);  
    v[8] = new Galeon( name: "Galeon1", tank_capacity: "700L", Owner: "Mateusz", max_weight: 10000, Jolly_Roger: "Czaszka Piracka");  
    //Wyswietlanie w petli informacji o kazdym obiekcie zapisanym w tablicy  
    for (int i = 0; i < v.length; i++){  
        v[i].Info();  
    }  
}
```

The Run window shows the output of the program:

```
Wlasciciel : Jaroslaw. Quad o nazwie: Quad1. Pojemnosc silnika :100L. Wyciagajacy maks predkosc: 80km/h. 0 silniku: tlokowy. Typ amortyzatorow: terenowe  
Wlasciciel : Mateusz. Motocykl o nazwie: motor1. Pojemnosc baku: 50L. Maksymalna predkosc: 50km/h. Typ silnika: tlokowy. Wysokosc zawieszenia: 10cm  
Wlasciciel : Michal. Samochod o nazwie: Auto1. Pojemnosc baku: 100L. Maksymalna predkosc: 150km/h. Typ silnika: tlokowy. Maksymalna liczba pasazerow: 5  
Wlasciciel : Jakub. Samolot o nazwie: Samolot1. Pojemnosc baku: 1000L. Maksymalna wysokosc lotu: 2000m.n.p.m. Typ samolotu: Odrzutowy  
Wlasciciel : Filip. Zaglowka o nazwie: Zaglowka1. Pojemnosc baku: 200L. Maksymalny dozvolony ciezar na pokladzie: 5000kg. Liczba zagli: 2  
Wlasciciel : Maksym. Samolot o nazwie: Helikopten1. Pojemnosc baku: 500L. Maksymalna wysokosc lotu: 1000m.n.p.m. Dlugosc smigiel helikoptera: 8m  
Wlasciciel : Kamil. Samolot o nazwie: Balon1. Pojemnosc baku: 300kg. Maksymalna wysokosc lotu: 800m.n.p.m. Sila ognia w balonie: 70  
Wlasciciel : Marcel. Zaglowka o nazwie: Podwodna1. Pojemnosc baku: 800L. Maksymalny dozvolony ciezar na pokladzie: 10000kg. Maksymalne zanuzenie pod woda: 50  
Wlasciciel : Mateusz. Zaglowka o nazwie: Galeon1. Pojemnosc baku: 700L. Maksymalny dozvolony ciezar na pokladzie: 10000kg. Flaga statku: Czaszka Piracka
```