


```
package Kontroler.Kontroler;

import Widok.Widok.*;

public class Kontroler {

    private WyświetlanieInformacji wyświetlenieInformacji;
    private Kontekst kontekst;
    private ObsługaLinii obsługaLinii;
    private ObsługaBiletów obsługaBiletów;
    private FasadaInterakcji fasadaInterakcji;

    /**
     *
     * @param nrBiletu
     */
    public boolean sprawdzWaznoscBiletu(int nrBiletu) {
        // TODO - implement Kontroler.sprawdzWaznoscBiletu
        return False;
    }

    /**
     *
     * @param nrLinii
     */
    public void dodajLinie(int nrLinii) {
        // TODO - implement Kontroler.dodajLinie
    }

    public boolean zatwierdzZgodnoscOsoby() {
        // TODO - implement Kontroler.zatwierdzZgodnoscOsoby
        return False;
    }

    public int podajNrBiletu() {
        // TODO - implement Kontroler.podajNrBiletu
        return NrBiletu;
    }

    public int podajNrLinii() {
        // TODO - implement Kontroler.podajNrLinii
        return NrLinii;
    }

    public void podajGodzineOdjazdu() {
        // TODO - implement Kontroler.podajGodzineOdjazdu
    }

    public void podajNazwePrzystanku() {
        // TODO - implement Kontroler.podajNazwePrzystanku
    }

    public void podajNrRejesstracyjny() {
        // TODO - implement Kontroler.podajNrRejesstracyjny
    }

    /**
     *
     * @param args
     */
}
```

```
public static void main(String[] args) {  
    // TODO - implement Kontroler.main  
}  
  
}
```

```
package Kontroler.Kontroler;  
  
public interface ObslugaLinii {  
  
    /**  
     *  
     * @param nrLinii  
     */  
    void dodajLinie(int nrLinii);  
  
}
```

```
package Kontroler.Kontroler;  
  
public interface ObslugaBiletow {  
  
    /**  
     *  
     * @param nrBiletu  
     */  
    boolean sprawdzWaznosc(int nrBiletu);  
  
}
```

```
package Kontroler.Kontroler;  
  
import Widok.Widok.*;  
  
public class FasadaInterakcji implements InterakcjeZUzytkownikiem {  
  
    public int podajNrBiletu() {  
        // TODO - implement FasadaInterakcji.podajNrBiletu  
    }  
  
    public boolean zatwierdzZgodnoscOsoby() {  
        // TODO - implement FasadaInterakcji.zatwierdzZgodnoscOsoby  
    }  
  
    public int podajNrLinii() {  
        // TODO - implement FasadaInterakcji.podajNrLinii  
    }  
  
    public void podajNazwePrzystanku() {  
        // TODO - implement FasadaInterakcji.podajNazwePrzystanku  
    }  
  
    public void podajNrRejesstracyjny() {  
        // TODO - implement FasadaInterakcji.podajNrRejesstracyjny  
    }  
  
    public void podajGodzineOdjazdu() {  
        // TODO - implement FasadaInterakcji.podajGodzineOdjazdu  
    }  
  
}
```

```
}  
}
```

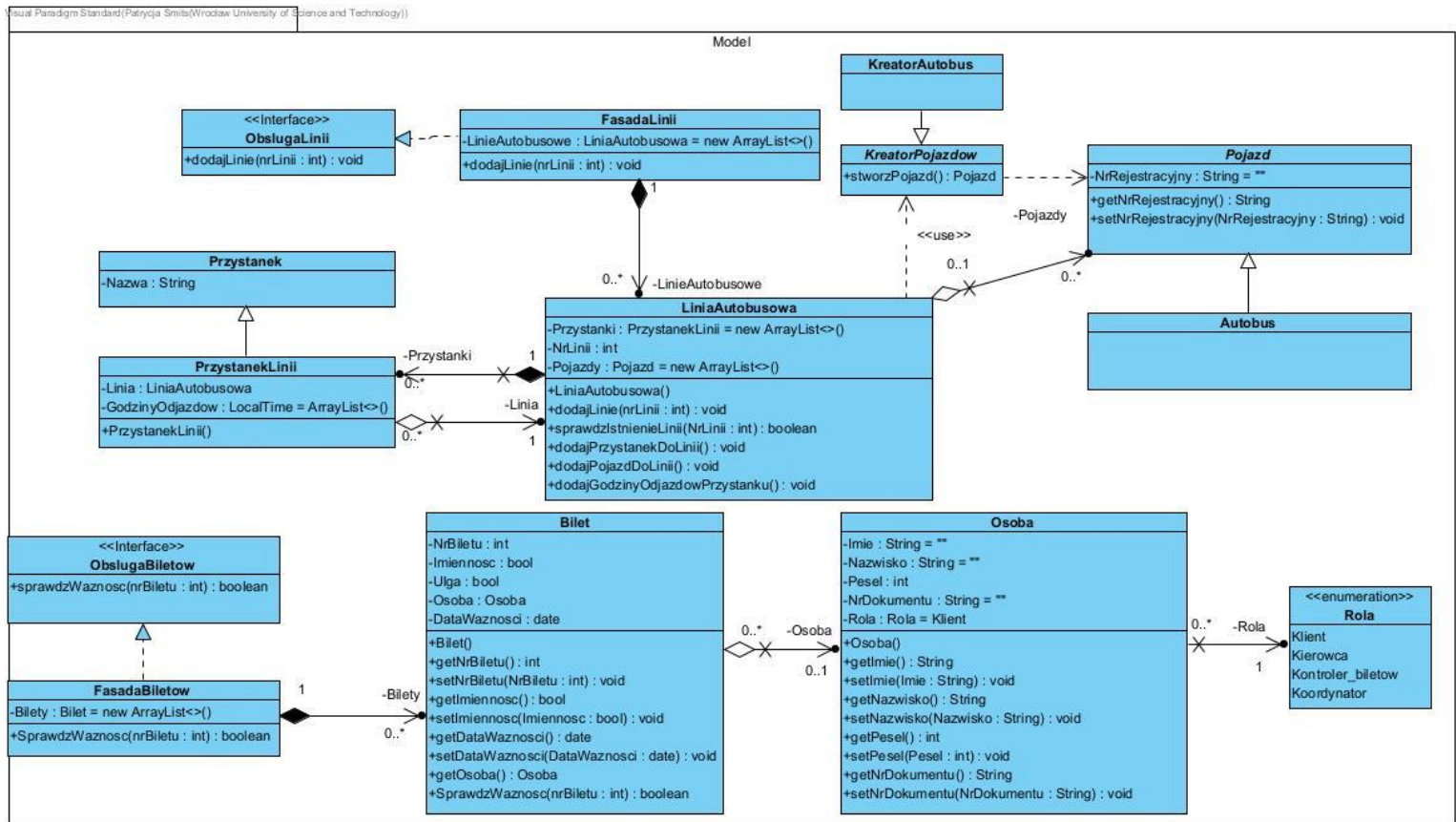
```
package Kontroler.Kontroler;  
  
public class Kontekst {  
  
    private StrategiaSprawdzaniaBiletow strategia;  
  
    /**  
     *  
     * @param strategia  
     */  
    public void setStrategia(StrategiaSprawdzaniaBiletow strategia) {  
        this.strategia = strategia;  
    }  
  
    public void wykonajStrategie() {  
        // TODO - implement Kontekst.wykonajStrategie  
    }  
}
```

```
package Kontroler.Kontroler;  
  
public interface StrategiaSprawdzaniaBiletow {  
  
    void wykonajStrategie();  
}
```

```
package Kontroler.Kontroler;  
  
public class StrategiaSprawdzaniaKlienta implements  
StrategiaSprawdzaniaBiletow {  
  
    public void wykonajStrategie() {  
        // TODO - implement StrategiaSprawdzaniaKlienta.wykonajStrategie  
    }  
}
```

```
package Kontroler.Kontroler;  
  
public class StrategiaSprawdzaniaKontrolera implements  
StrategiaSprawdzaniaBiletow {  
  
    public void wykonajStrategie() {  
        // TODO - implement StrategiaSprawdzaniaKontrolera.wykonajStrategie  
    }  
}
```

3. Diagram klas warstwy modelu



```

package Model.Model;

public class Bilet {

    private int NrBiletu;
    private bool Imiennosc;
    private bool Ulg;
    private Osoba Osoba;
    private date DataWaznosci;

    public Bilet() {
        // TODO - implement Bilet.Bilet
    }

    public int getNrBiletu() {
        // TODO - implement Bilet.getNrBiletu
    }

    /**
     *
     * @param NrBiletu
     */
    public void setNrBiletu(int NrBiletu) {
        // TODO - implement Bilet.setNrBiletu
    }

    public bool getImiennosc() {
        // TODO - implement Bilet.getImiennosc
    }
}

```

```

/**
 *
 * @param Imiennosc
 */
public void setImiennosc(bool Imiennosc) {
    // TODO - implement Bilet.setImiennosc
}

public date getDataWaznosci() {
    // TODO - implement Bilet.getDataWaznosci
}

/**
 *
 * @param DataWaznosci
 */
public void setDataWaznosci(date DataWaznosci) {
    // TODO - implement Bilet.setDataWaznosci
}

public Osoba getOsoba() {
    // TODO - implement Bilet.getOsoba
}

/**
 *
 * @param nrBiletu
 */
public boolean SprawdzWaznosc(int nrBiletu) {
    // TODO - implement Bilet.SprawdzWaznosc
}
}

```

```

package Model.Model;

import Kontroler.Kontroler.*;
import java.util.*;

public class FasadaBiletow implements ObslugaBiletow {

    private Collection<Bilet> Bilety = new ArrayList<>();

    /**
     *
     * @param nrBiletu
     */
    public boolean SprawdzWaznosc(int nrBiletu) {
        // TODO - implement FasadaBiletow.SprawdzWaznosc
    }
}

```

```

package Model.Model;

public class Osoba {

    private String Imie = "";
}

```

```

private String Nazwisko = "";
private int Pesel;
private String NrDokumentu = "";
private Rola Rola = Model.Model.Rola.Klient;

public Osoba() {
    // TODO - implement Osoba.Osoba
}

public String getImie() {
    // TODO - implement Osoba.getImie
}

/**
 *
 * @param Imie
 */
public void setImie(String Imie) {
    // TODO - implement Osoba.setImie
}

public String getNazwisko() {
    // TODO - implement Osoba.getNazwisko
}

/**
 *
 * @param Nazwisko
 */
public void setNazwisko(String Nazwisko) {
    // TODO - implement Osoba.setNazwisko
}

public int getPesel() {
    // TODO - implement Osoba.getPesel
}

/**
 *
 * @param Pesel
 */
public void setPesel(int Pesel) {
    // TODO - implement Osoba.setPesel
}

public String getNrDokumentu() {
    // TODO - implement Osoba.getNrDokumentu
}

/**
 *
 * @param NrDokumentu
 */
public void setNrDokumentu(String NrDokumentu) {
    // TODO - implement Osoba.setNrDokumentu
}
}

```

```
package Model.Model;
```

```
public enum Rola {  
    Klient,  
    Kierowca,  
    Kontroler_biletow,  
    Koordynator  
}
```

```
package Model.Model;  
  
public abstract class Pojazd {  
  
    private String NrRejestracyjny = "";  
  
    public String getNrRejestracyjny() {  
        // TODO - implement Pojazd.getNrRejestracyjny  
    }  
  
    /**  
     *  
     * @param NrRejestracyjny  
     */  
    public void setNrRejestracyjny(String NrRejestracyjny) {  
        // TODO - implement Pojazd.setNrRejestracyjny  
    }  
}
```

```
package Model.Model;  
  
public abstract class KreatorPojazdow {  
  
    public Pojazd stworzPojazd() {  
        // TODO - implement KreatorPojazdow.stworzPojazd  
    }  
}
```

```
package Model.Model;  
  
public abstract class KreatorPojazdow {  
  
    public Pojazd stworzPojazd() {  
        // TODO - implement KreatorPojazdow.stworzPojazd  
    }  
}
```

```
package Model.Model;  
  
public class Autobus extends Pojazd {  
}
```

```
package Model.Model;  
  
import java.util.*;
```



```

public class LiniaAutobusowa {

    private Collection<PrzystanekLinii> Przystanki = new ArrayList<>();
    private int NrLinii;
    private Collection<Pojazd> Pojazdy = new ArrayList<>();

    public LiniaAutobusowa() {
        // TODO - implement LiniaAutobusowa.LiniaAutobusowa
        throw new UnsupportedOperationException();
    }

    /**
     *
     * @param nrLinii
     */
    public void dodajLinie(int nrLinii) {
        // TODO - implement LiniaAutobusowa.dodajLinie
    }

    /**
     *
     * @param NrLinii
     */
    public boolean sprawdzIstnienieLinii(int NrLinii) {
        // TODO - implement LiniaAutobusowa.sprawdzIstnienieLinii
    }

    public void dodajPrzystanekDoLinii() {
        // TODO - implement LiniaAutobusowa.dodajPrzystanekDoLinii
    }

    public void dodajPojazdDoLinii() {
        // TODO - implement LiniaAutobusowa.dodajPojazdDoLinii
    }

    public void dodajGodzinyOdjazdowPrzystanku() {
        // TODO - implement LiniaAutobusowa.dodajGodzinyOdjazdowPrzystanku
    }

}

```

```

package Model.Model;

import java.util.*;

public class FasadaLinii {

    private Collection<LiniaAutobusowa> LinieAutobusowe = new ArrayList<>();

    /**
     *
     * @param nrLinii
     */
    public void dodajLinie(int nrLinii) {
        // TODO - implement FasadaLinii.dodajLinie
    }

}

```

```
package Model.Model;

public class Przystanek {

    private String Nazwa;

}
```

```
package Model.Model;

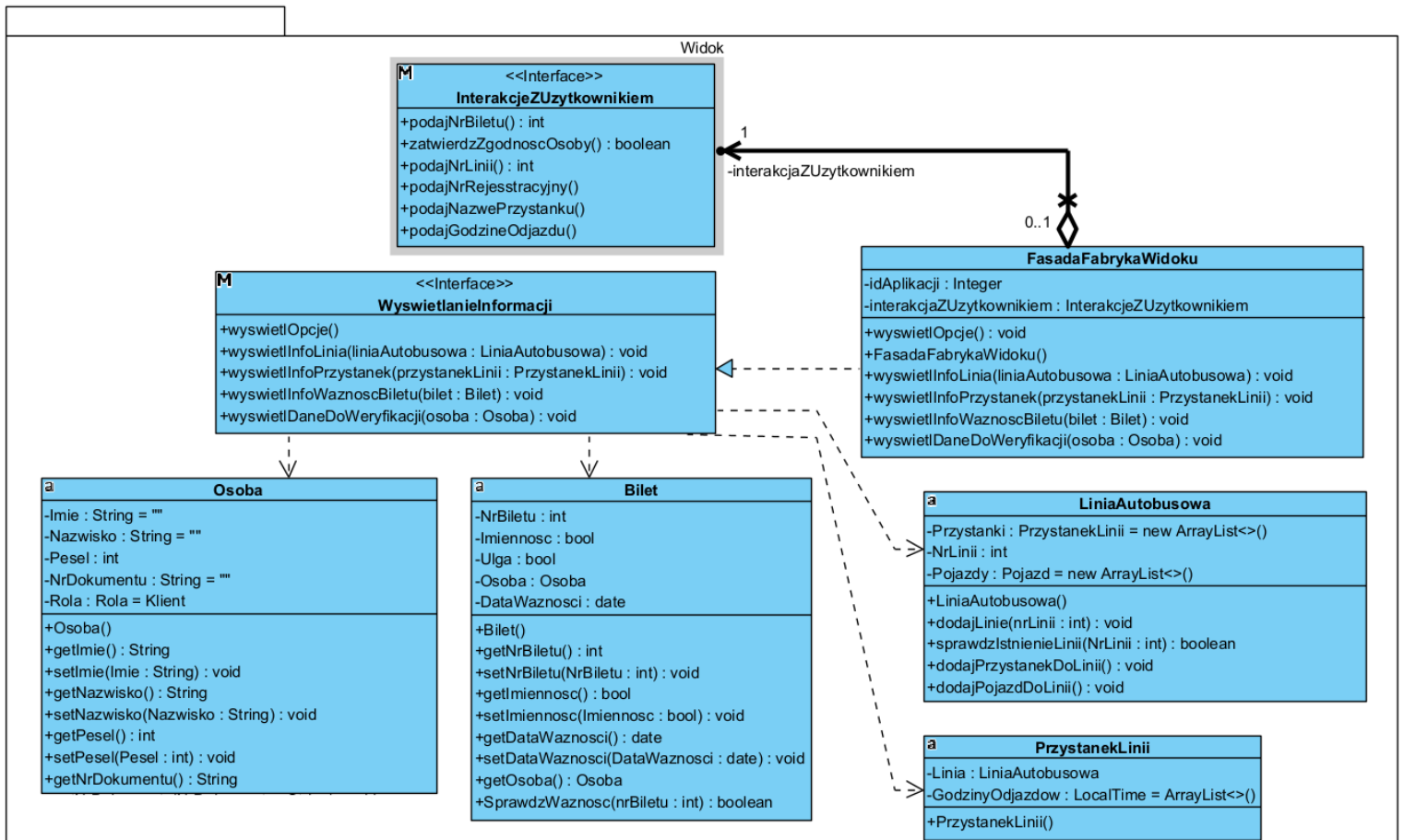
public class PrzystanekLinii extends Przystanek {

    private LiniaAutobusowa Linia;
    private LocalTime GodzinyOdjazdow = ArrayList<>();

    public PrzystanekLinii() {
        // TODO - implement PrzystanekLinii. PrzystanekLinii
    }

}
```

4. Diagram klas warstwy prezentacji (widok)



```

package Widok.Widok;

import java.util.*;
import Model.Model.*;

public class FasadaFabrykaWidoku {

    private Collection<Integer> idAplikacji;
    private InterakcjeZUzytkownikiem interakcjaZUzytkownikiem;

    public void wyswietlOpcje() {
        // TODO - implement FasadaFabrykaWidoku.wyswietlOpcje
    }

    public FasadaFabrykaWidoku() {
        // TODO - implement FasadaFabrykaWidoku.FasadaFabrykaWidoku
    }

    /**
     *
     * @param liniaAutobusowa
     */
    public void wyswietlInfoLinia(LiniaAutobusowa liniaAutobusowa) {
        // TODO - implement FasadaFabrykaWidoku.wyswietlInfoLinia
    }

    /**

```

```

    *
    * @param przystanekLinii
    */
    public void wyswietlInfoPrzystanek(PrzystanekLinii przystanekLinii) {
        // TODO - implement FasadaFabrykaWidoku.wyswietlInfoPrzystanek
    }

    /**
     *
     * @param bilet
     */
    public void wyswietlInfoWaznoscBiletu(Bilet bilet) {
        // TODO - implement FasadaFabrykaWidoku.wyswietlInfoWaznoscBiletu
    }

    /**
     *
     * @param osoba
     */
    public void wyswietlDaneDoWeryfikacji(Osoba osoba) {
        // TODO - implement FasadaFabrykaWidoku.wyswietlDaneDoWeryfikacji
    }
}

```

```

package Widok.Widok;

public interface InterakcjeZUzytkownikiem {

    int podajNrBiletu();

    boolean zatwierdzZgodnoscOsoby();

    int podajNrLinii();

    void podajNrRejesstracyjny();

    void podajNazwePrzystanku();

    void podajGodzineOdjazdu();

}

```

```

package Widok.Widok;

import Model.Model.*;

public interface WyświetlanieInformacji {

    void wyswietlOpcje();

    /**
     *
     * @param liniaAutobusowa
     */
    void wyswietlInfoLinia(LiniaAutobusowa liniaAutobusowa);

    /**
     *
     * @param przystanekLinii
     */
}

```

```

    */
    void wyswietlInfoPrzystanek(PrzystanekLinii przystanekLinii);

    /**
     *
     * @param bilet
     */
    void wyswietlInfoWaznoscBiletu(Bilet bilet);

    /**
     *
     * @param osoba
     */
    void wyswietlDaneDoWeryfikacji(Osoba osoba);
}

```

5. Uzasadnienie wykorzystanych wzorców projektowych

W projekcie wykorzystane zostały wzorce projektowe w celu ułatwienia organizacji kodu zgodnie z założeniami programowania obiektowego SOLID. Kod implementuje model warstwowy w architekturze MVC.

W warstwie kontrolera zastosowane zostały wzorce:

- **FasadaInterakcji**, która upraszcza interakcje z użytkownikiem, umożliwiając kontrolerowi korzystanie z jednego punktu dostępu do metod wymagających interakcji. Dodatkowo fasada zwiększa czytelność kodu i ułatwia zarządzanie systemem.
- Wzorec **Strategia** umożliwia definiowanie różnych algorytmów w oddzielnych klasach, a następnie ich wymienne stosowanie. Klasa **Kontekst** zarządza wyborem odpowiedniej strategii, umożliwiając dynamiczne określenie logiki sprawdzania biletów. Klasy **StrategiaSprawdzaniaKlienta** i **StrategiaSprawdzaniaKontrolera** pozwalają na implementację różnych procedur sprawdzania biletów w zależności od roli użytkownika. Dzięki temu kod jest elastyczny i łatwo rozszerzalny w przypadku dodania nowych strategii.

W warstwie modelu zastosowane zostały wzorce:

- Wzorec Fabryka Abstrakcyjna umożliwia tworzenie obiektów bez ujawniania ich konkretnej klasy. **KreatorPojazdow** definiuje ogólną metodę tworzenia pojazdów, a konkretne klasy pojazdów np. Autobus, są tworzone za pomocą fabryki. Wzorec Fabryka został zaimplementowany, ponieważ ułatwia dodawanie nowych typów pojazdów w przyszłości bez modyfikacji istniejącego kodu.
- Wzorec Fasada został wykorzystany w klasach FasadaBiletow oraz **FasadaLinii** w celu uproszczenia obsługi klas Bilety oraz LiniaAutobusowa. Dzięki temu zarządzanie biletami oraz liniami autobusowymi odbywać się będzie w sposób scentralizowany, bez ujawniania wewnętrznych szczegółów implementacji.

W warstwie widoku zastosowany został wzorec Fasada w klasie **FasadaFabrykaWidoku**, który jednocześnie jest Fabryką odpowiedzialną za tworzenie jego elementów. Pełni rolę centralnego punktu dostępu do operacji związanych z wyświetlaniem informacji i interakcjami użytkownika. Dzięki temu Kontroler komunikuje się z systemem widoku poprzez tylko jedną klasę, co zmniejsza

jego złożoność i odpowiedzialność, a w przyszłości umożliwi prostsze modyfikacje sposobu zarządzania widokiem.