

Jakub Biadalski 253857

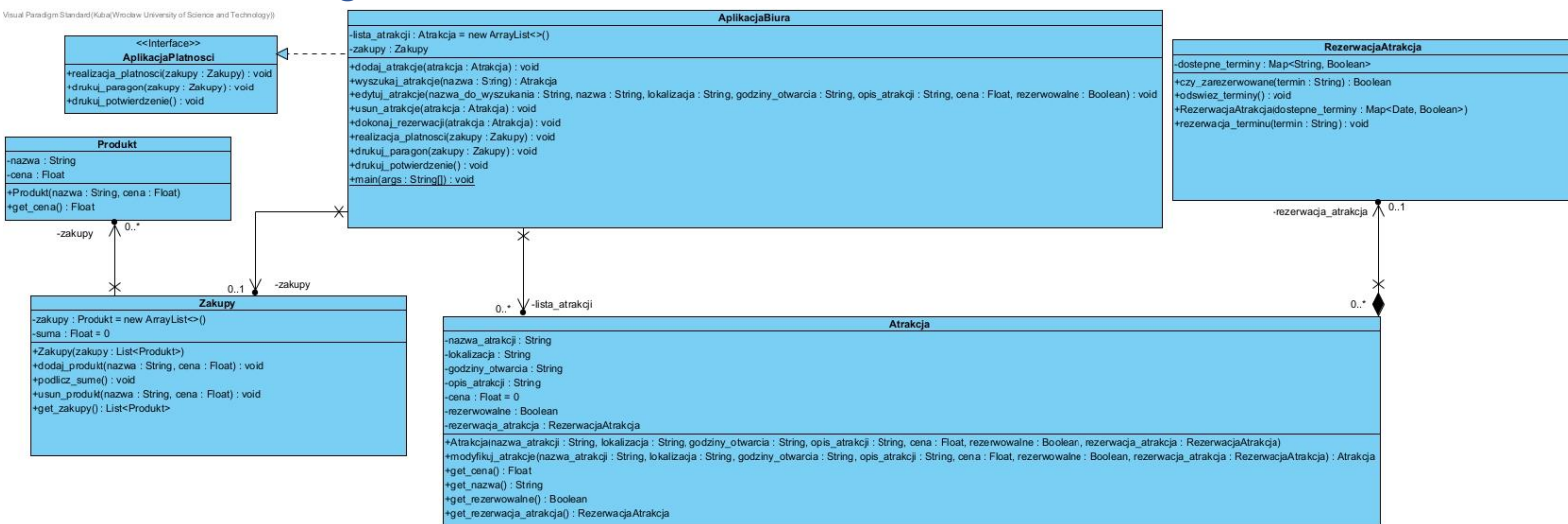
Michał Jarzab 263980

Paweł Cyganiuk 263983

# Laboratorium 7-10 - Diagram klas i diagramy sekwencji

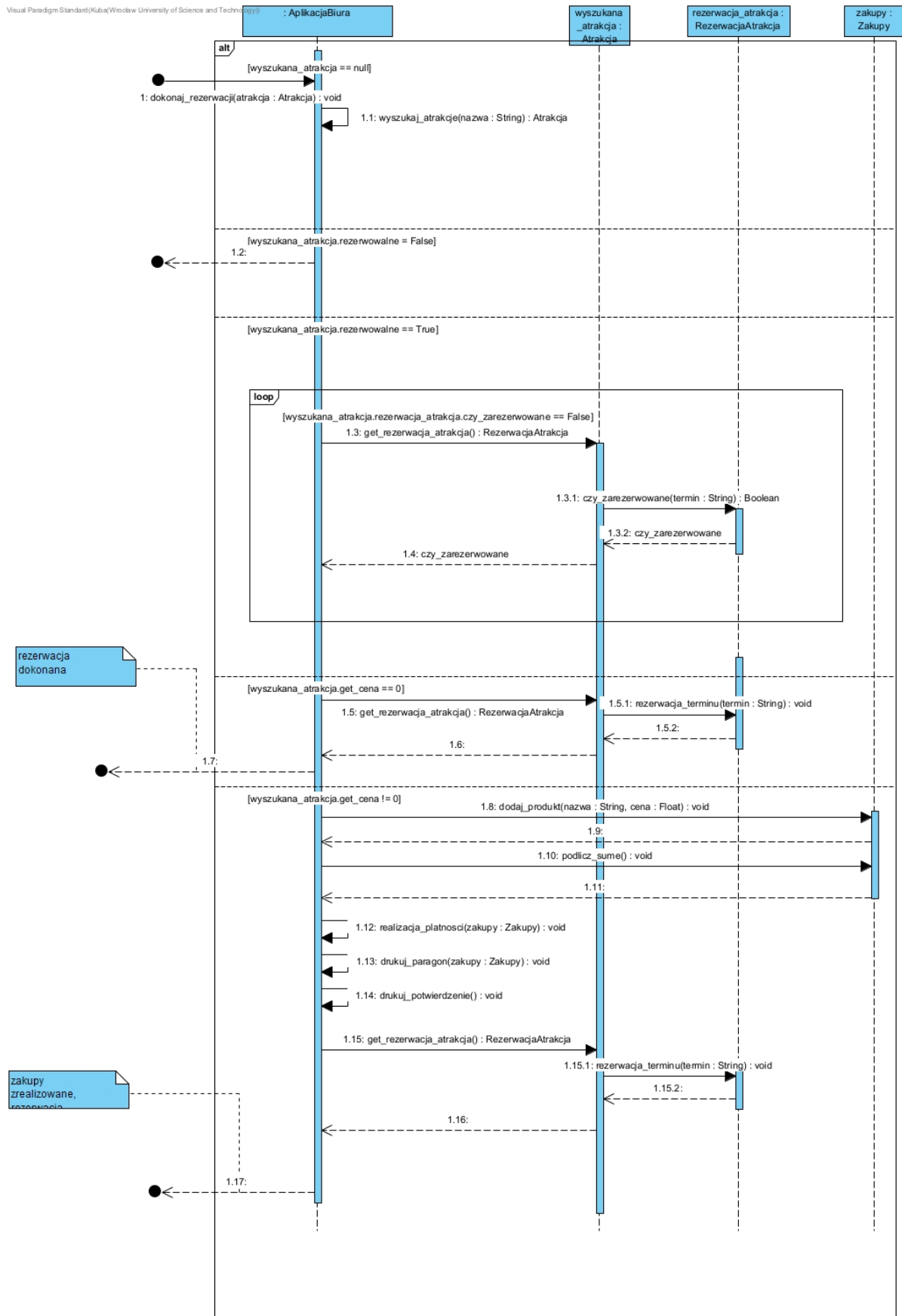
## 1. Diagram klas

Visual Paradigm Standard (Kuba/Wrocław University of Science and Technology)

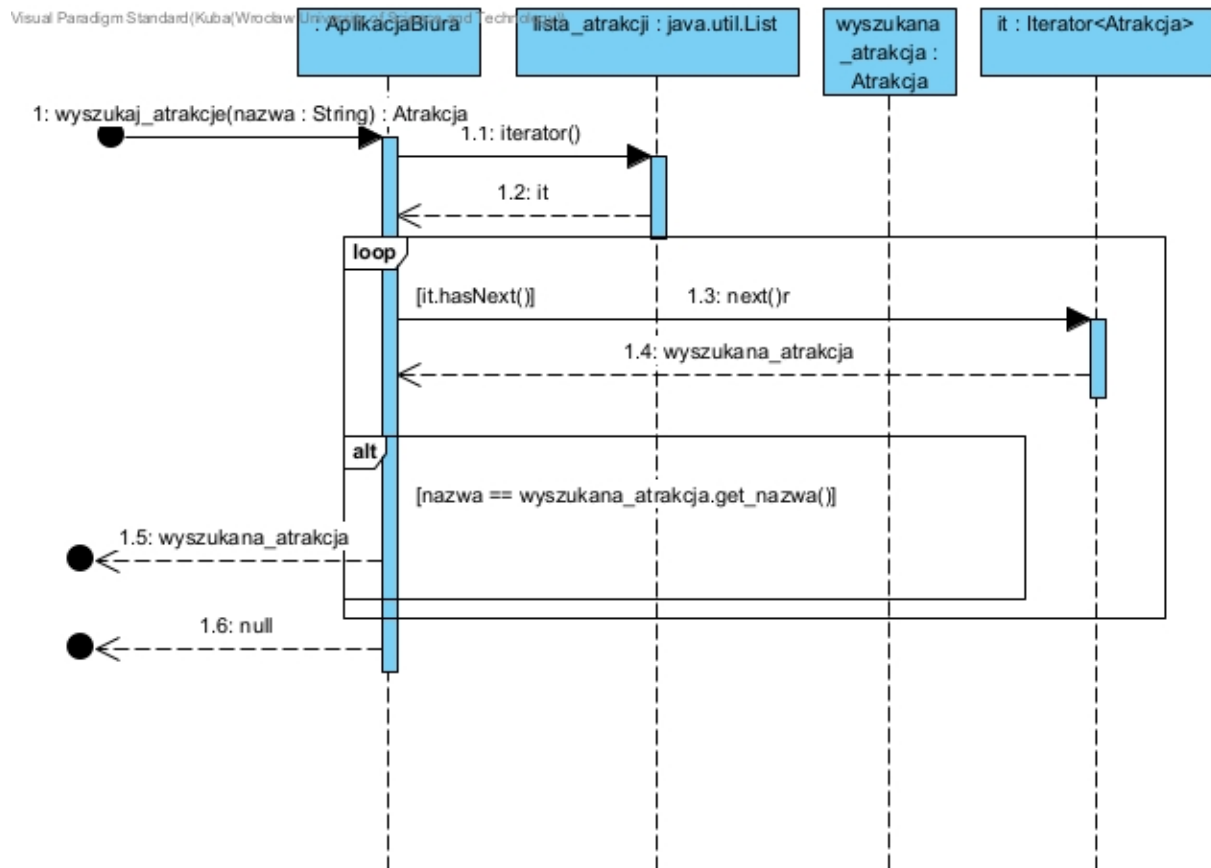


## 2. Diagram sekwencji

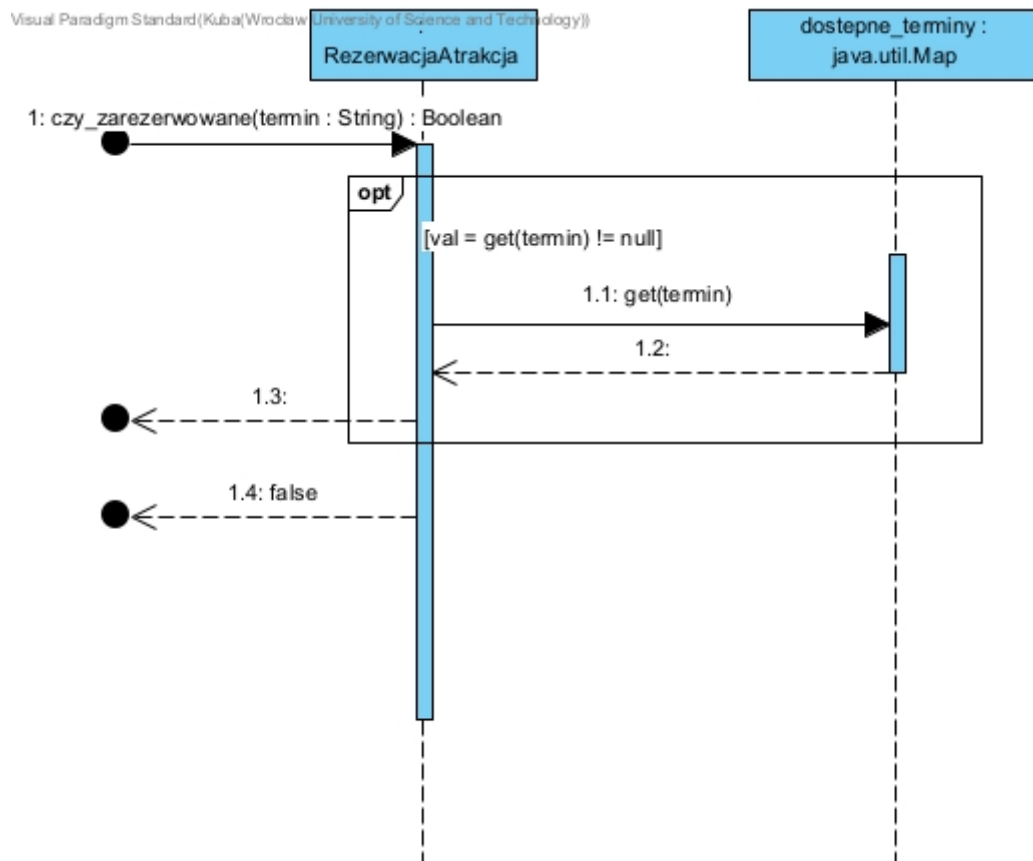
### 2.1. Diagram sekwencji metody dokonaj\_rezerwacji



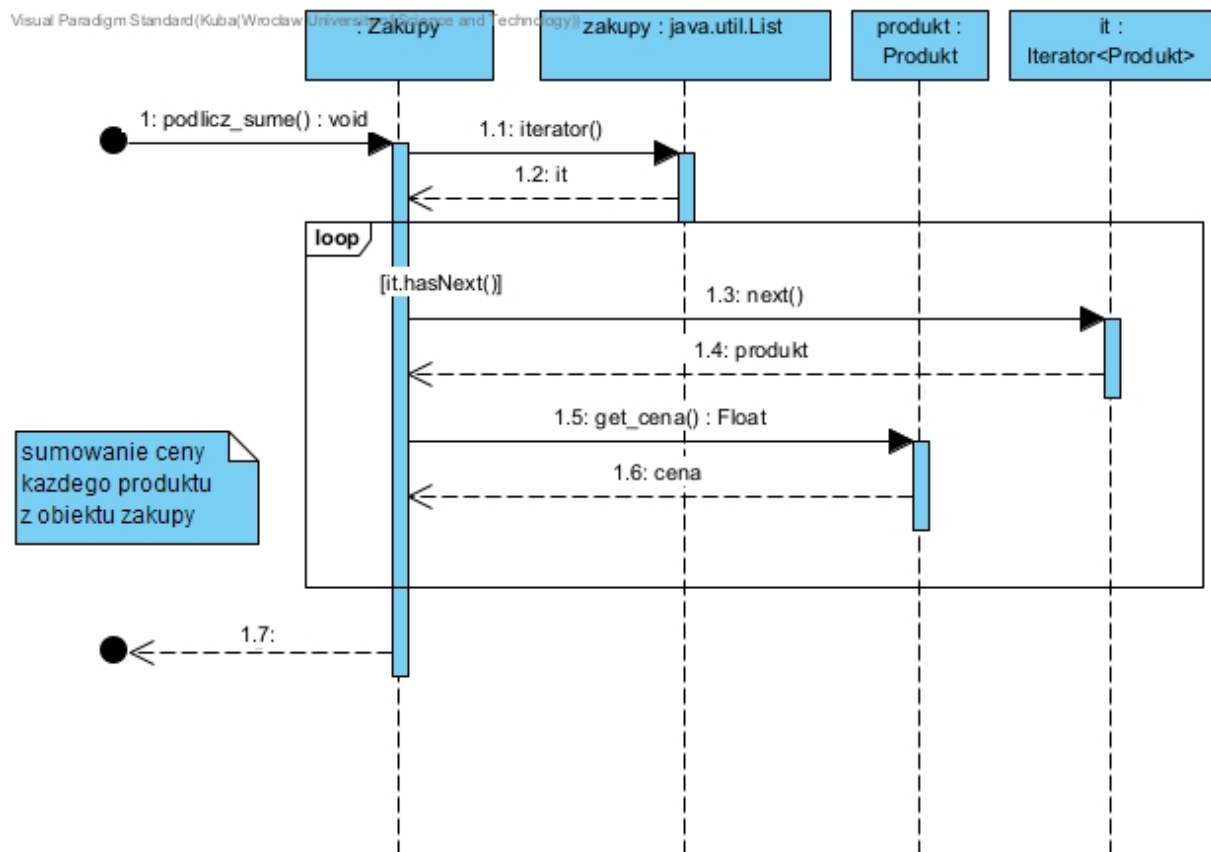
## 2.2. Diagram sekwencji metody wyszukaj\_atrakcje



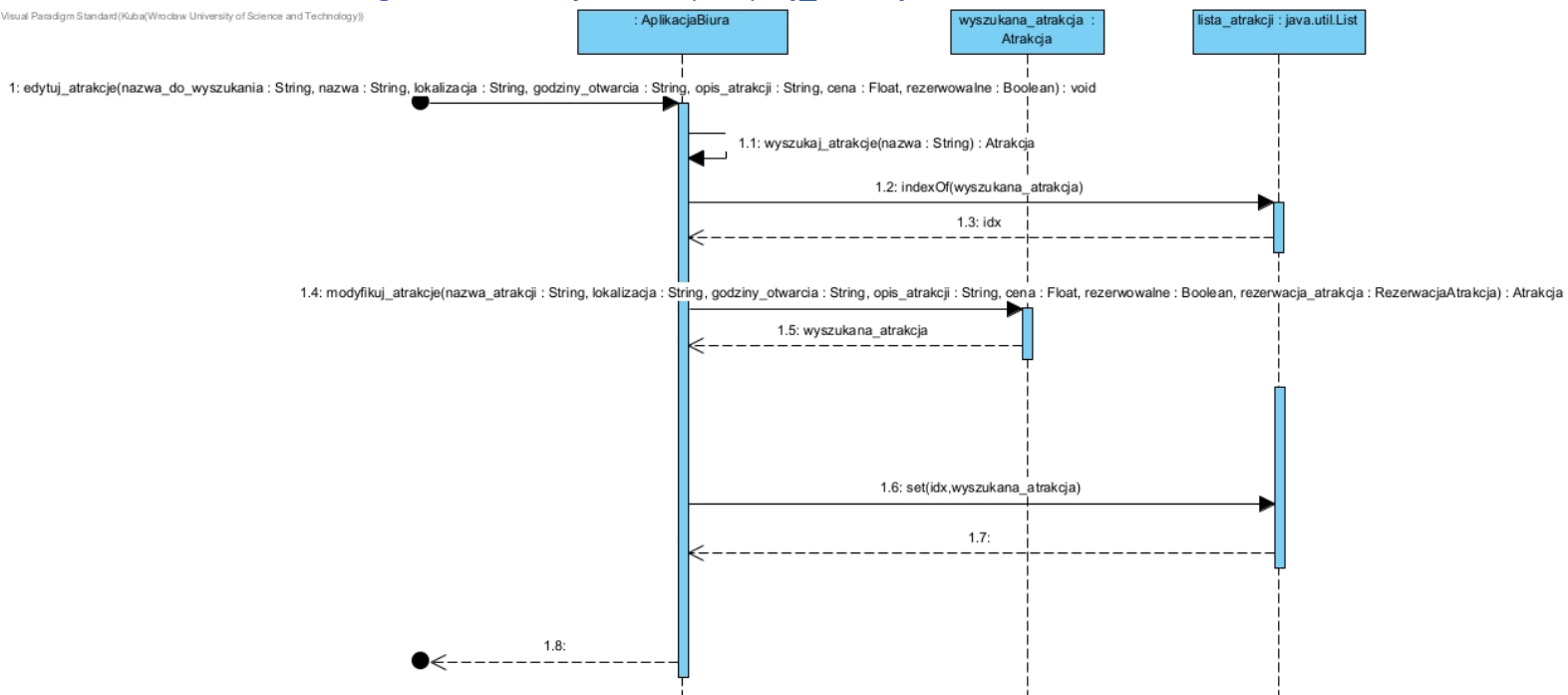
## 2.3. Diagram sekwencji metody czy\_zarezerwowane



## 2.4. Diagram sekwencji metody podlicz\_sume



## 2.5. Diagram sekwencji metody edytuj\_atrakcje



### 3. Kody Klas

#### 3.1. Kod klasy Produkt

```
public class Produkt {  
  
    private String nazwa;  
    private Float cena;  
  
    public Produkt(String nazwa, Float cena) {  
        this.nazwa = nazwa;  
        this.cena = cena;  
    }  
    public Float get_cena(){  
        return this.cena;  
    }  
  
}
```

#### 3.2. Kod klasy Zakupy

```
import java.util.Iterator;  
import java.util.List;  
  
public class Zakupy {  
  
    private List<Produkt> zakupy;  
    private Float suma = 0F;  
  
    public Zakupy(List<Produkt> zakupy) {  
        this.zakupy = zakupy;  
    }  
  
    public void dodaj_produkt(String nazwa, Float cena) {  
        zakupy.add(new Produkt(nazwa, cena));  
    }  
  
    public void podlicz_sume() {  
        Produkt produkt;  
        Iterator<Produkt> it = zakupy.iterator();  
        while (it.hasNext()) {  
            produkt = it.next();  
            this.suma += produkt.get_cena();  
        }  
    }  
    public void usun_produkt(String nazwa, Float cena) {  
        zakupy.remove(new Produkt(nazwa, cena));  
    }  
  
    public List<Produkt> get_zakupy() {  
        return this.zakupy;  
    }  
  
}
```

#### 3.3. Kod klasy AplikacjaPlatnosci

```
public interface AplikacjaPlatnosci {  
  
    void realizacja_platnosci(Zakupy zakupy);  
    void drukuj_paragon(Zakupy zakupy);  
    void drukuj_potwierdzenie();  
  
}
```

### 3.4. Kod klasy Atrakcja

```
public class Atrakcja {

    private String nazwa_atrakcji;
    private String lokalizacja;
    private String godziny_otwarcia;
    private String opis_atrakcji;
    private Float cena = 0F;
    private Boolean rezerwowalne;
    private RezerwacjaAtrakcja rezerwacja_atrakcja;

    public Atrakcja(String nazwa_atrakcji, String lokalizacja, String
godziny_otwarcia, String opis_atrakcji, Float cena, Boolean
rezerwowalne, RezerwacjaAtrakcja rezerwacja_atrakcja) {
        this.nazwa_atrakcji = nazwa_atrakcji;
        this.lokalizacja = lokalizacja;
        this.godziny_otwarcia = godziny_otwarcia;
        this.opis_atrakcji = opis_atrakcji;
        this.cena = cena;
        this.rezerwowalne = rezerwowalne;
        this.rezerwacja_atrakcja = rezerwacja_atrakcja;
    }

    public Atrakcja modyfikuj_atrakcje(String nazwa_atrakcji, String
lokalizacja, String godziny_otwarcia, String opis_atrakcji, Float
cena, Boolean rezerwowalne, RezerwacjaAtrakcja rezerwacja_atrakcja) {
        if(nazwa_atrakcji != null) this.nazwa_atrakcji =
nazwa_atrakcji;
        if(lokalizacja != null) this.lokalizacja = lokalizacja;
        if(godziny_otwarcia != null) this.godziny_otwarcia =
godziny_otwarcia;
        if(opis_atrakcji != null) this.opis_atrakcji = opis_atrakcji;
        if(cena != null) this.cena = cena;
        if(rezerwowalne != null) this.rezerwowalne = rezerwowalne;
        if(rezerwacja_atrakcja != null) this.rezerwacja_atrakcja =
rezerwacja_atrakcja;
        return this;
    }

    public Float get_cena() {
        return this.cena;
    }

    public String get_nazwa() {
        return this.nazwa_atrakcji;
    }

    public Boolean get_rezerwowalne() {
        return this.rezerwowalne;
    }

    public RezerwacjaAtrakcja get_rezerwacja_atrakcja() {
        return this.rezerwacja_atrakcja;
    }
}
```

### 3.5. Kod klasy RezerwacjaAtrakcja

```
import java.util.Map;

public class RezerwacjaAtrakcja {

    private Map<String, Boolean> dostepne_terminy;

    public RezerwacjaAtrakcja (Map<String, Boolean> dostepne_terminy)
    {
        this.dostepne_terminy = dostepne_terminy;
    }

    public Boolean czy_zarezerwowane(String termin) {
        if (dostepne_terminy.get(termin) != null){
            return dostepne_terminy.get(termin);
        }
        return false;
    }

    public void odswiez_terminy() {
        System.out.println("Odswiezanie terminów");
    }

    public void rezerwacja_terminu(String termin) {
        this.dostepne_terminy.replace(termin, Boolean.FALSE);
    }

}
```

### 3.6. Kod klasy AplikacjaBiura

```
import java.util.Iterator;
import java.util.List;
import java.util.Scanner;
public class AplikacjaBiura implements AplikacjaPlatnosci {

    private List<Atrakcja> lista_atrakcji;
    private Zakupy zakupy;

    public void dodaj_atrakcje(Atrakcja atrakcja) {
        this.lista_atrakcji.add(atrakcja);
    }

    public Atrakcja wyszukaj_atrakcje(String nazwa) {
        Atrakcja wyszukana_atrakcja;
        Iterator<Atrakcja> it = lista_atrakcji.iterator();
        while (it.hasNext()){
            wyszukana_atrakcja = it.next();
            if(nazwa == wyszukana_atrakcja.get_nazwa()) return
wyszukana_atrakcja;
        }
        System.out.println("Nie znaleziono atrakcji.");
        return null;
    }

    public void edytuj_atrakcje(String nazwa_do_szukania,String nazwa,
String lokalizacja, String godziny_otwarcia, String opis_atrakcji,
Float cena, Boolean rezerwowalne) {
        Atrakcja wyszukana_atrakcja =
```

```

wyszukaj_atrakcje(nazwa_do_szukania);
    int idx = this.lista_atrakcji.indexOf(wyszukana_atrakcja);
    wyszukana_atrakcja.modyfikuj_atrakcje(nazwa, lokalizacja,
godziny_otwarcia, opis_atrakcji, cena, rezerwowalne, null);
    this.lista_atrakcji.set(idx, wyszukana_atrakcja);
}

public void usun_atrakcje(Atrakcja atrakcja) {
    this.lista_atrakcji.remove(atrakcja);
}

public void dokonaj_rezerwacji(Atrakcja atrakcja, String termin) {
    Scanner sc = new Scanner(System.in);
    Atrakcja wyszukana_atrakcja =
this.wyszukaj_atrakcje(atrakcja.get_nazwa());
    if(wyszukana_atrakcja.get_rezerwowalne() == false) return;
    if(wyszukana_atrakcja.get_rezerwowalne() == true){

while(wyszukana_atrakcja.get_rezerwacja_atrakcja().czy_zarezerwowane(
termin) == Boolean.FALSE){

        System.out.println("Podaj inny termin(Format RR:MM:DD
GG:MinMin): ");
        termin = sc.next();

    }
    if(wyszukana_atrakcja.get_cena() == 0){

wyszukana_atrakcja.get_rezerwacja_atrakcja().rezerwacja_terminu(termi
n);

        System.out.println("Rezerwacja dokonana");
        return;
    }
    else{
        zakupy.dodaj_produkt(wyszukana_atrakcja.get_nazwa(),
wyszukana_atrakcja.get_cena());
        zakupy.podlicz_sume();
        realizacja_platnosci(zakupy);
        drukuj_paragon(zakupy);
        drukuj_potwierdzenie();

wyszukana_atrakcja.get_rezerwacja_atrakcja().rezerwacja_terminu(termi
n);

        System.out.println("Rezerwacja dokonana");
        return;
    }
}

}

public void realizacja_platnosci(Zakupy zakupy) {
    System.out.println("Metoda realizacja platnosci wykonana!");
}

public void drukuj_paragon(Zakupy zakupy) {
    System.out.println("Metoda drukuj paragon wykonana!");
}

public void drukuj_potwierdzenie() {
    System.out.println("Metoda drukuj potwierdzenie wykonana!");
}

```



```
public static void main(String[] args) {  
    AplikacjaBiura aplikacja = new AplikacjaBiura();  
    Atrakcja atrakcja = null;  
}  
  
}
```