

Lista 2 | POO

Zadanie 1

1. (1p) Przedstawić diagram klas dla poniższego kodu:

```
public interface ICommand
{
    void Execute( string CommandName );
}

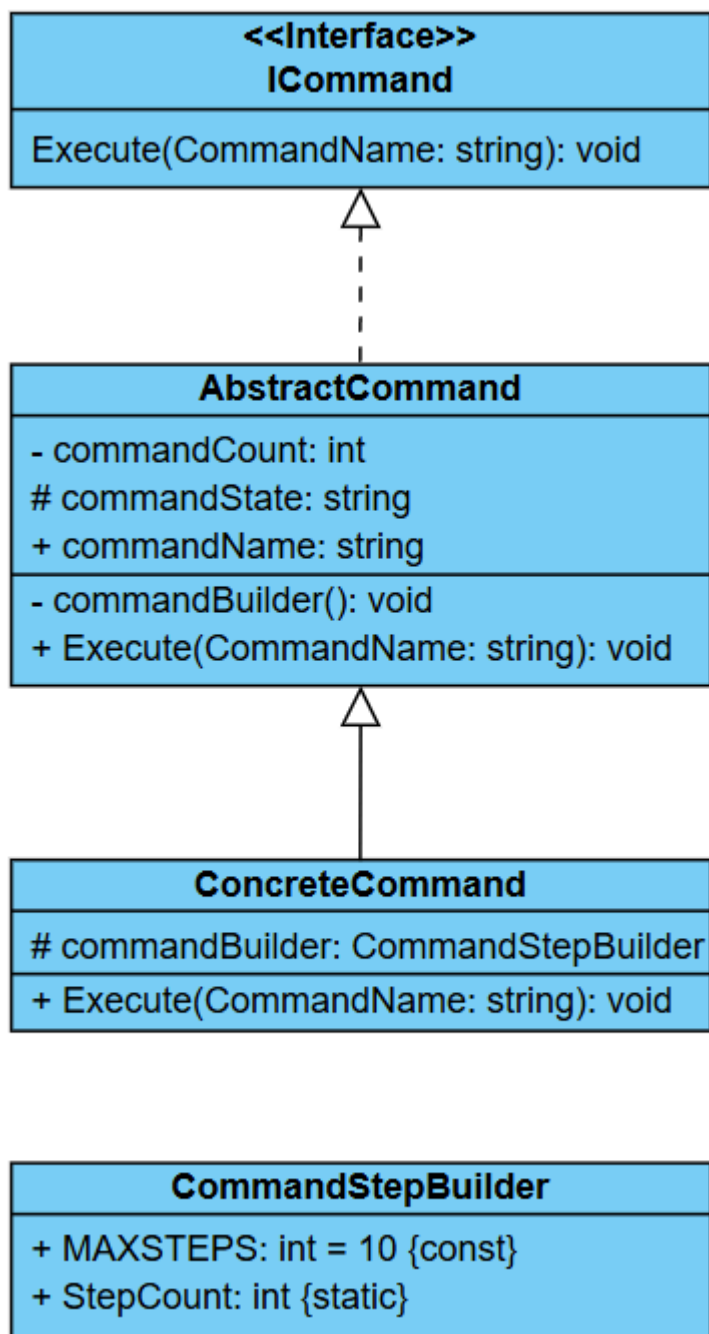
public abstract class AbstractCommand : ICommand
{
    private int commandCount;
    protected string commandState;
    public string commandName;

    private void commandBuilder() { }
    public abstract void Execute( string CommandName );
}

public class ConcreteCommand : AbstractCommand
{
    protected CommandStepBuilder commandBuilder;

    public override void Execute( string CommandName ) { }
}

public class CommandStepBuilder
{
    public const int MAXSTEPS = 10;
    public static int StepCount;
}
```

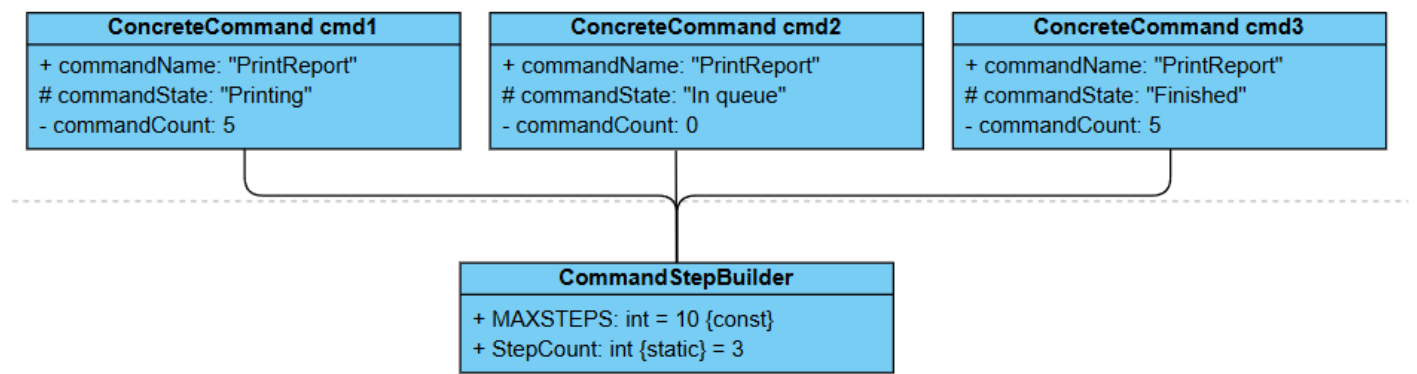


- Strzałka między `ConcreteCommand`, a `AbstractCommand` - dziedziczenie
- Strzałka między `AbstractCommand`, a `ICOMMAND` - implementacja interfejsu
- `ConcreteCommand` ma asocjacje do `CommandStepBuilder`

Zadanie 2

2. (1p) Przetawić przykładowy diagram obiektów dla klas z powyższego zadania.

Definicja: Diagram obiektów ma pokazać konkretne instancje i ich powiązania w danym momencie działania systemu.

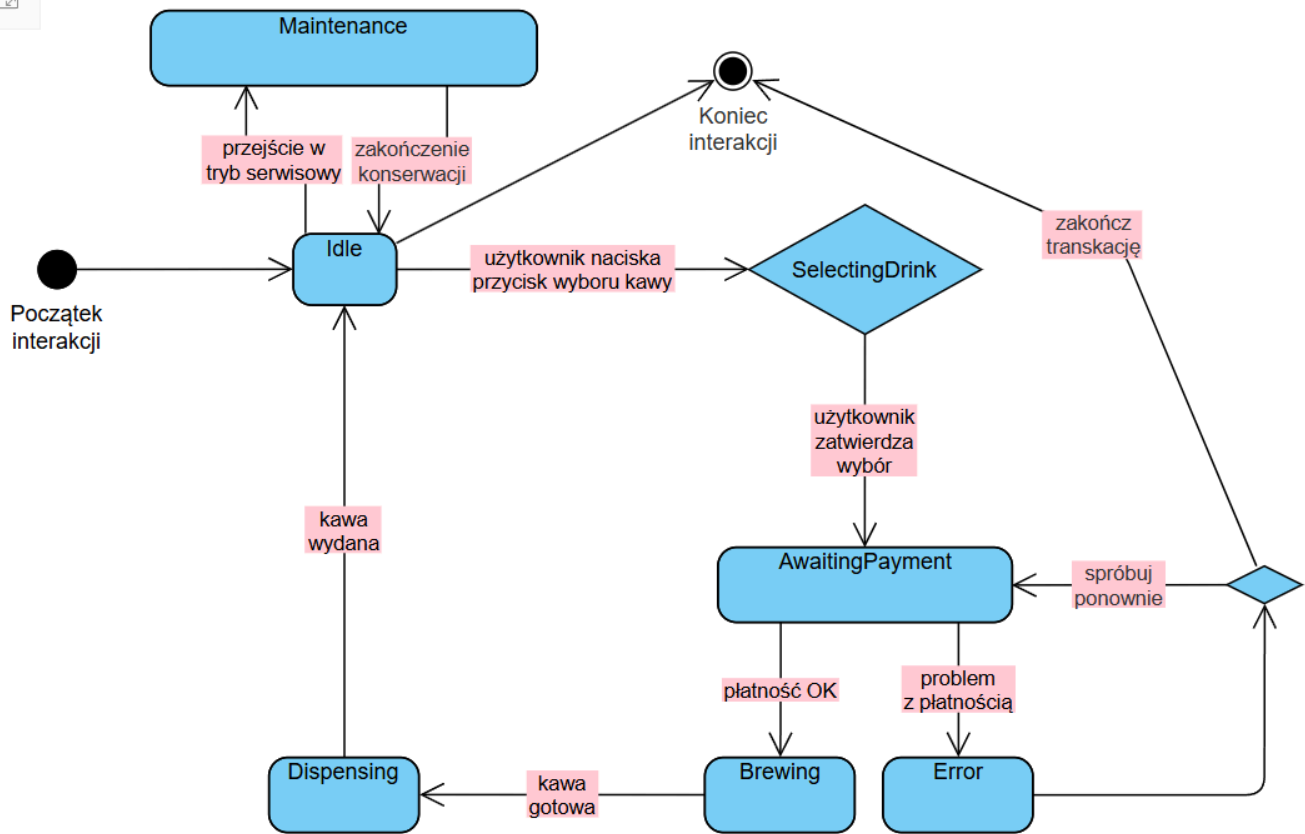


Zadanie 3

3. (1p) Przetwócić diagram stanów jakiegoś prostego urządzenia powszechnego użytku (bankomatu, automatu do kawy, automatu biletowego) (nie więcej niż kilkanaście stanów).

Urządzenie: Ekspres do kawy

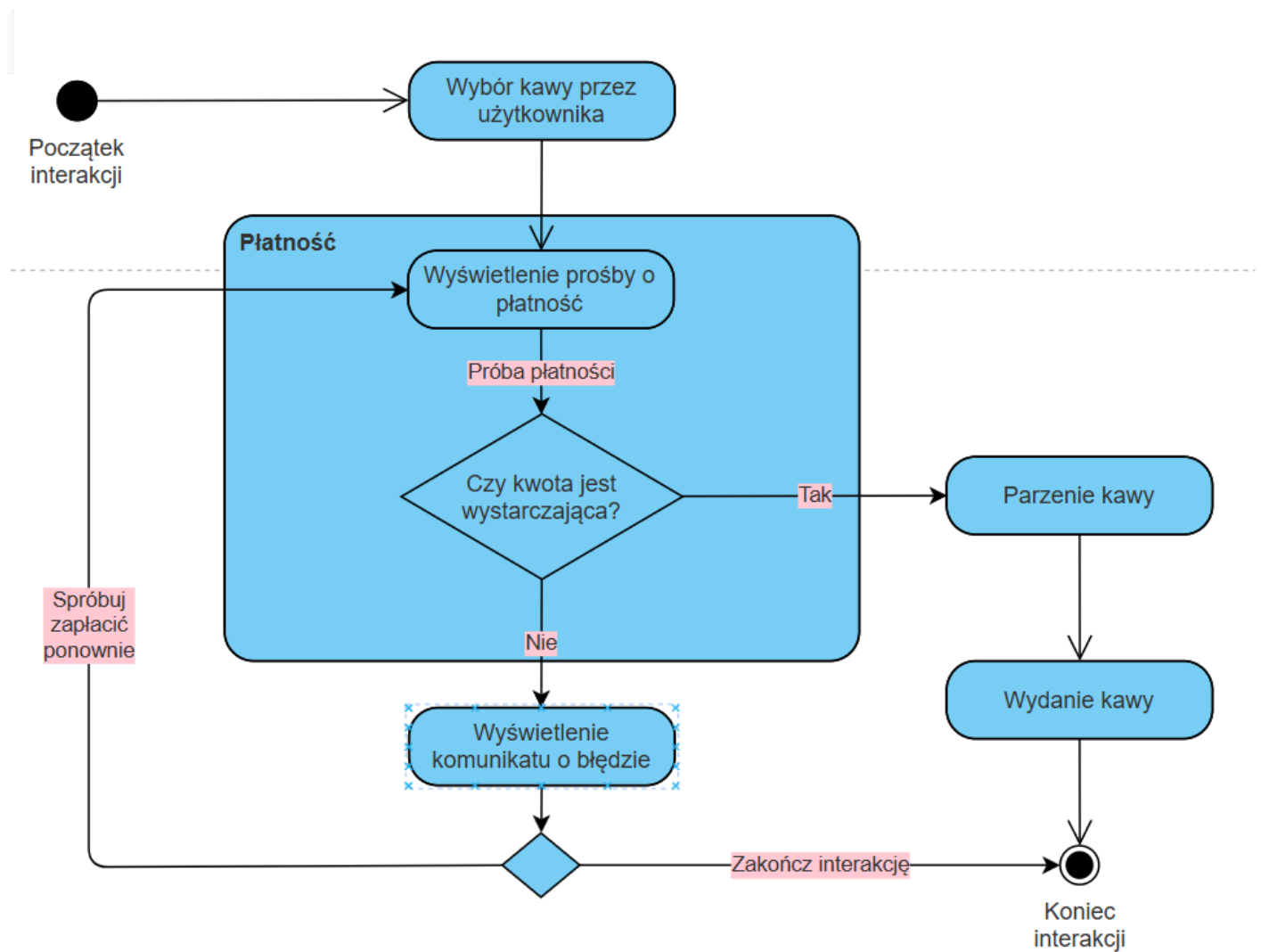
- **Idle** – ekspres czeka na działanie użytkownika
- **SelectingDrink** – użytkownik wybiera rodzaj kawy.
- **AwaitingPayment** – użytkownik musi wnieść opłatę.
- **Brewing** – ekspres przygotowuje kawę.
- **Dispensing** – ekspres wydaje gotowy napój do kubka.
- **Error** – np. brak wody, brak kawy, brak kubków itp.
- **Maintenance** – stan, w którym ekspres jest wyłączony z użytkowania i obsługiwany przez serwis.

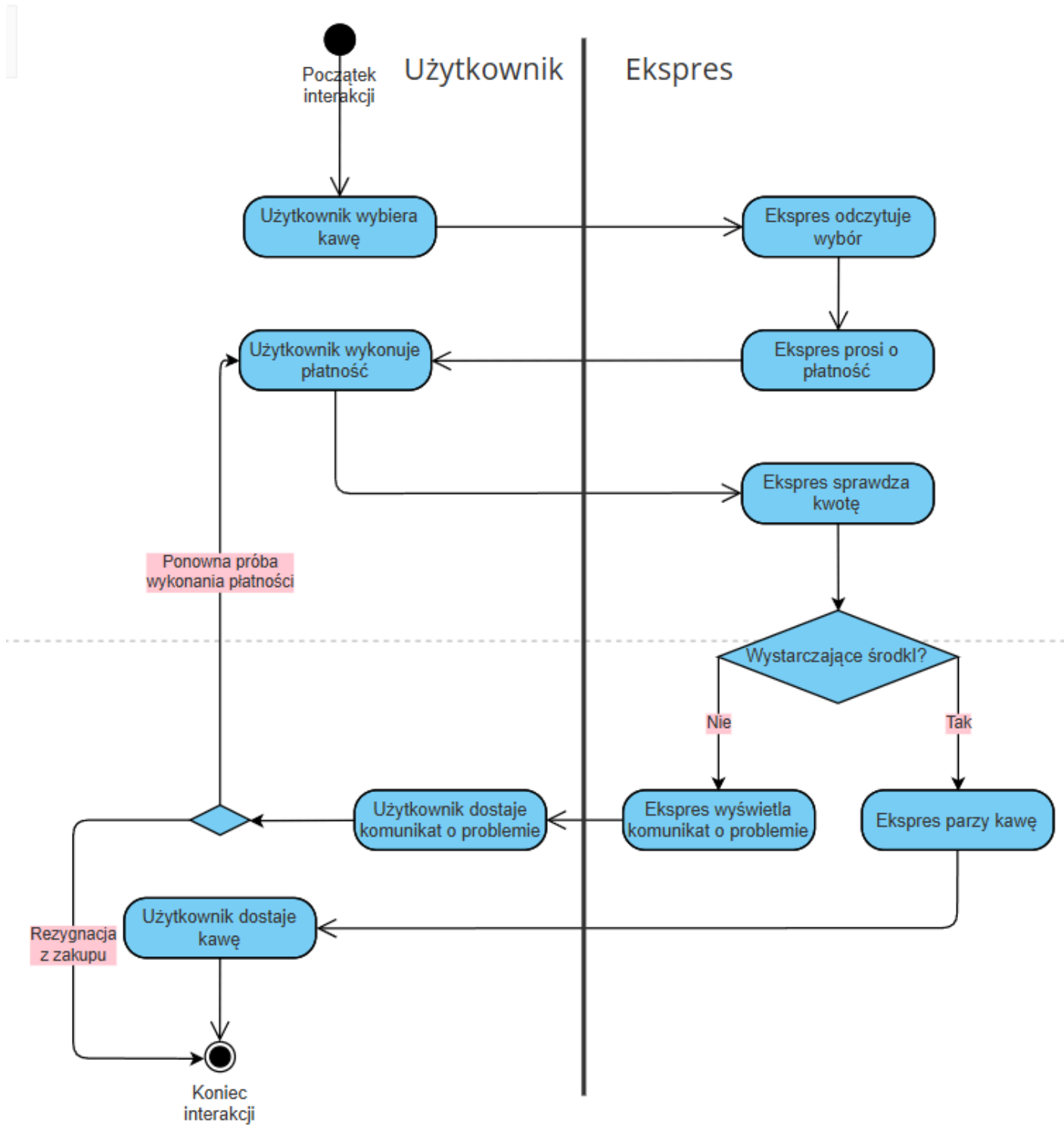


Zadanie 4

4. (2p) Przetwórz diagram czynności opisujących interakcję użytkownika z urządzeniem z poprzedniego zadania (nie więcej niż kilkanaście akcji). Przewidzieć jakieś sytuacje wyjątkowe (brak gotówki, błędnie wprowadzona kwota, błędny PIN), skutkujące pojawieniem się zdarzeń.

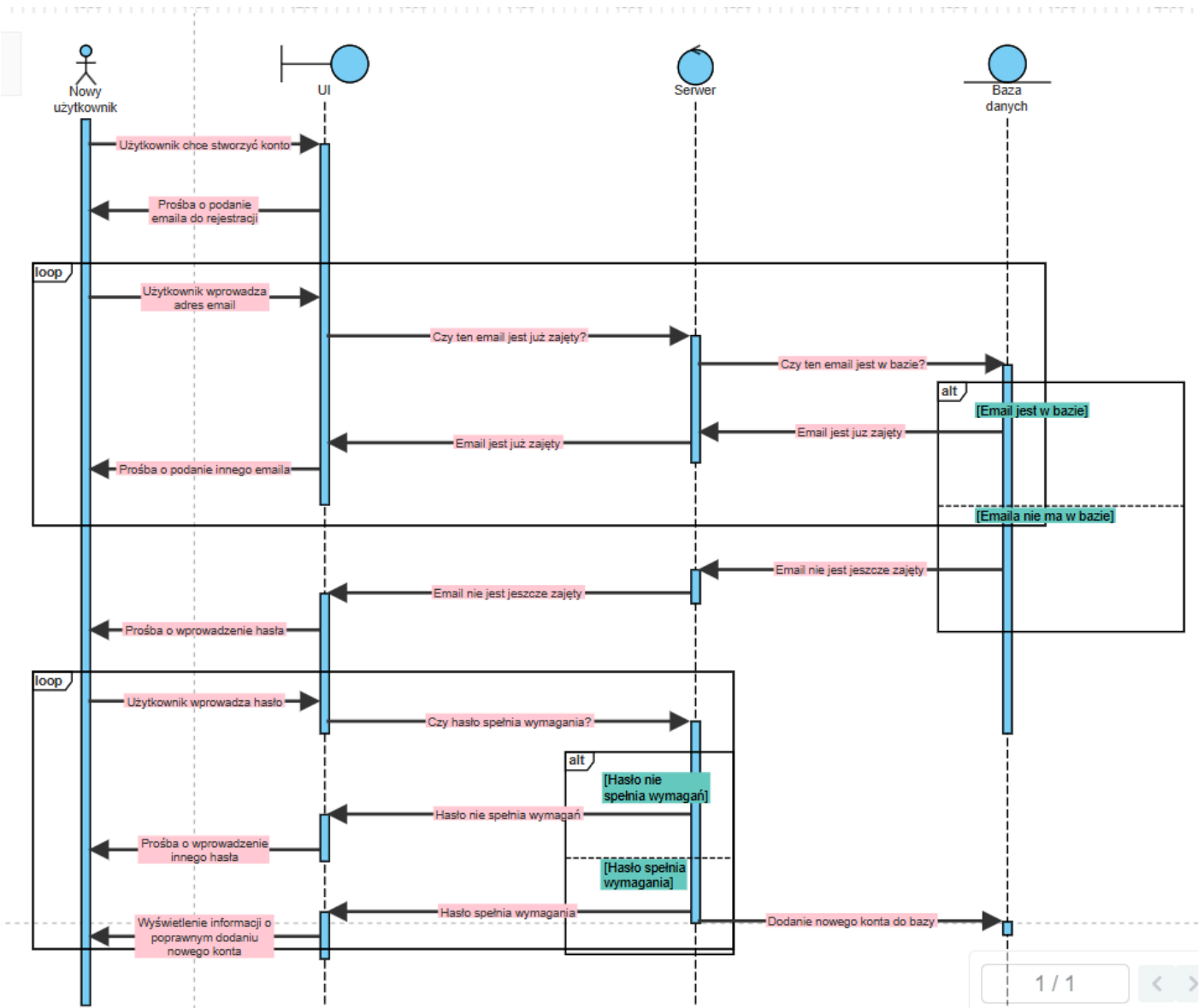
Przygotować diagram w wersji bez partycji, a następnie drugi - w wersji z co najmniej dwiema partycjami (użytkownik, urządzenie).





Zadanie 5

5. (2p) Zdokumentować w postaci diagramu sekwencji proces rejestracji nowego konta w przykładowej usłudze internetowej. Zdefiniować co najmniej użytkownika systemu i dwóch różnych uczestników procesu (różne typy odpowiedzialności): interfejs użytkownika i repozytorium danych.

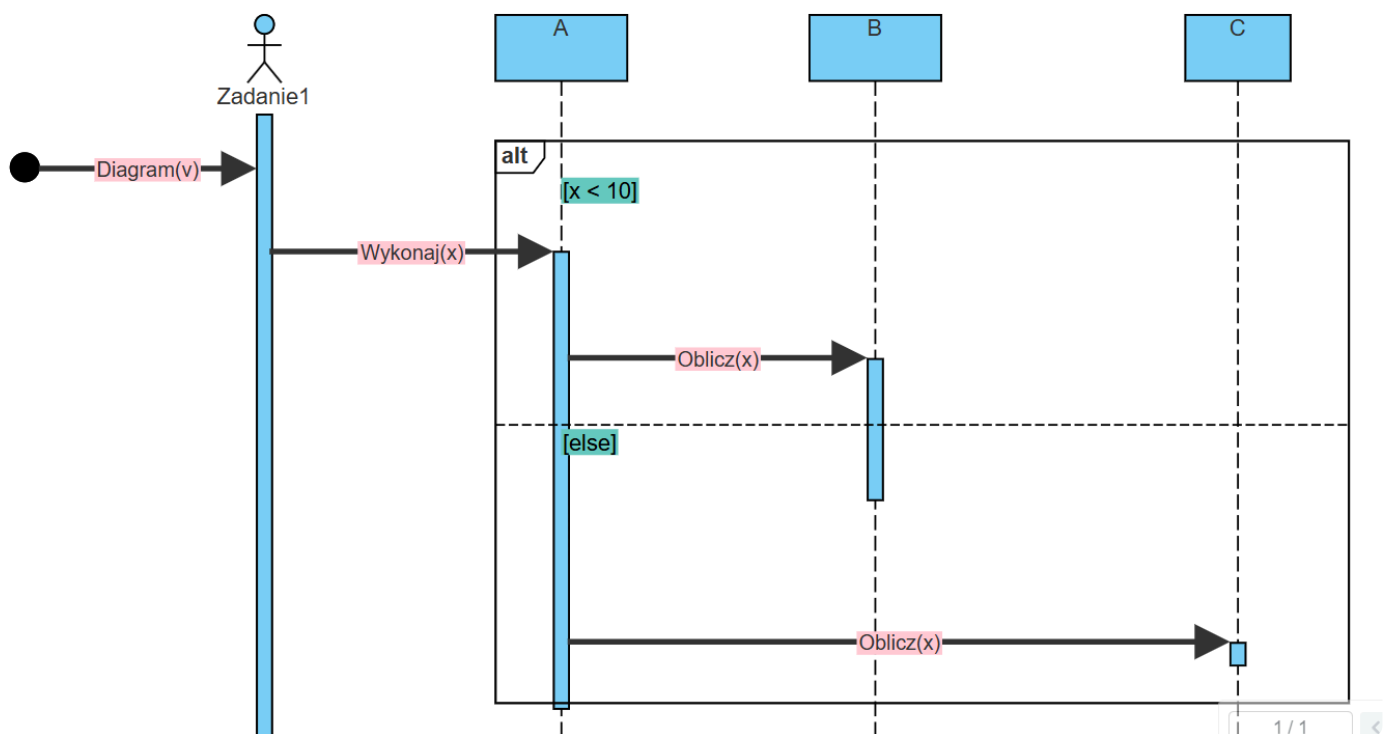


Zadanie 6

6. (1p) Przedstawić diagram sekwencji dla poniższego pseudokodu:

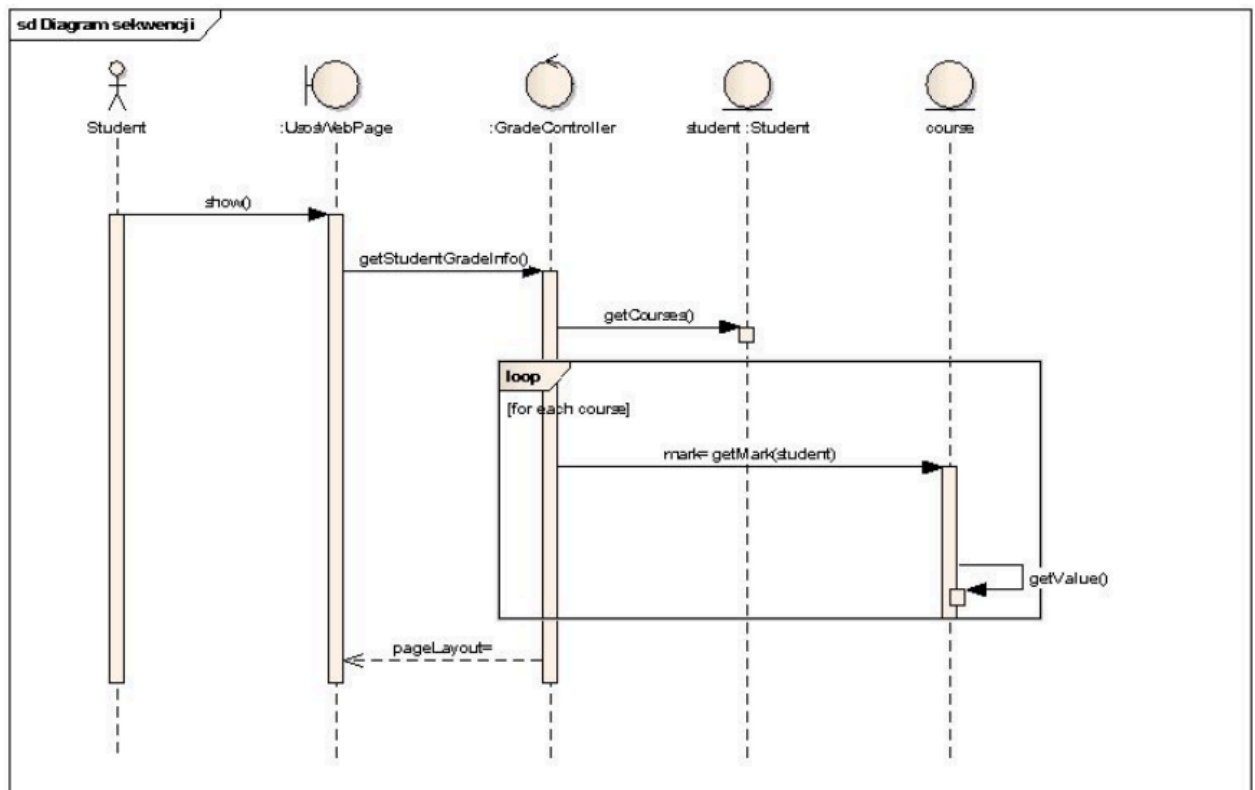
```
public class Zadanie1 {  
    A a;  
    public void Diagram( int v ) {  
        a.Wykonaj( v );  
    }  
}  
  
public class A {  
    B b; C c;  
    public void Wykonaj( int x ) {  
        if ( x < 10 ) {  
            b.Oblicz( x );  
        }  
        else  
            c.Oblicz( x );  
    }  
}  
  
public class B {  
    public void Oblicz( int n ) { }  
}  
  
public class C {  
    public void Oblicz( int m ) { }  
}
```

Czy na podstawie kodu można narysować jednoznaczny i precyzyjny diagram?



Zadanie 7

7. (1p) Napisać pseudokod dla diagramu sekwencji z rysunku 1.



Rysunek 1: Diagram sekwencji do zadania 7

Czy na podstawie diagramu można napisać jednoznaczny i precyzyjny kod?

```
class Student:
    indeksStudenta
    listaKursow studenta

    funkcja getCourses():
        zwróć listaKursow

class UsosWebPage:
    kontrolerOcen

    funkcja show():
        pageLayout = kontrolerOcen.getStudentGradeInfo()

class GradeController:
    student

    funkcja getStudentGradeInfo():
        listaOcen = pusta lista
        kursyStudenta = student.getCourses()

        dla kazdego kurs w kursyStudenta wykonaj:
            ocena = kurs.getMark(student)
            dodaj ocena do listaOcen

        zwróć listaOcen

class Course:
    listaOcenWKursie
    wybranyStudent

    funkcja getMark(student):
        wybranyStudent = student
        zwróć getValue()

    funkcja getValue():
        indeksStudenta = wybranyStudent.indeksStudenta
        zwróć listaOcenWKursie[indeksStudenta]
```

Zadanie 8

8. (1p) Wybrać któryś z diagramów z poprzednich zadań i odtworzyć w wybranym przez siebie narzędziu typu Diagrams-as-code (na przykład rekomendowane na wykładzie PlantUML czy Mermaid.js).

