

Projektowanie obiektowe oprogramowania

Zestaw 1

Język UML

2013-02-26

Liczba punktów do zdobycia: **10/10**

Zestaw ważny do: 2013-03-19

*Uwaga! W zadaniach w których mowa jest o przedstawieniu wybranego diagramu UML, należy użyć **jakiegoś** narzędzia typu CASE, ale niekoniecznie tego którego używano na wykładzie. Pośrednim celem zadań jest również bowiem zapoznanie się ze współczesnym warsztatem architekta oprogramowania - stąd wymaganie dedykowanego oprogramowania, a nie np. edytora graficznego w którym przy odrobinie wysiłku diagramy też można próbować rysować.*

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;
}
```

2. (**1p**) Przetawić przykładowy diagram obiektów dla klas z powyższego zadania.
3. (**1p**) Przetawić diagram stanów jakiegoś prostego urządzenia powszechnego użytku (bankomatu, automatu do kawy, automatu biletowego) (nie więcej niż kilkanaście stanów).
4. (**2p**) Przetawić 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).

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

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

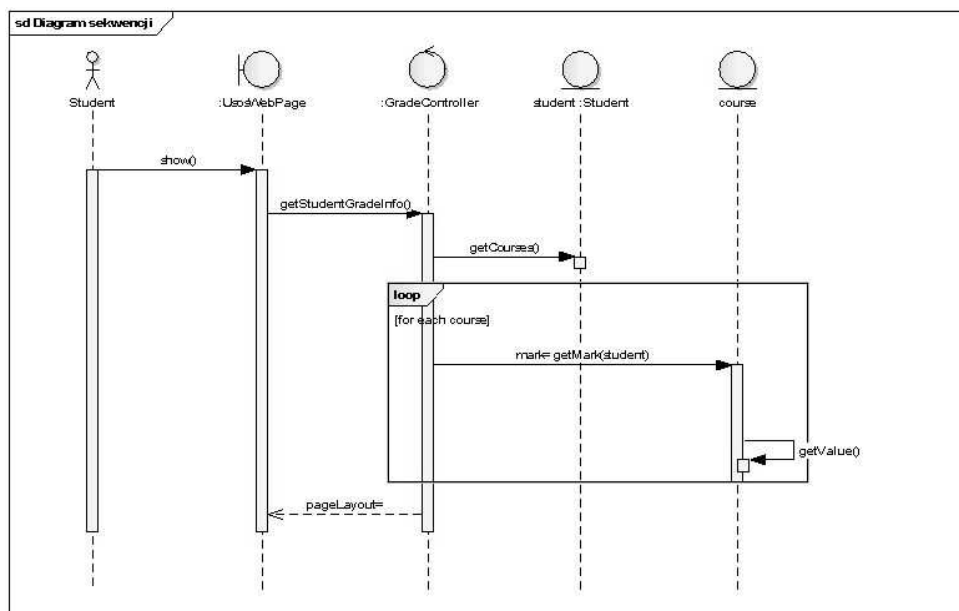
public class C {
    public void Oblicz( int m ) { }
}
```

Czy na podstawie kodu można narysować jednoznaczny i precyzyjny diagram?

7. (1p) Napisać pseudokod dla diagramu sekwencji z rysunku 1.
Czy na podstawie diagramu można napisać jednoznaczny i precyzyjny kod?
8. (1p) Narysować diagram czynności dla procesu Unified Process, w którym partycjami będą dyscypliny (zbieranie wymagań, projektowanie, wytwarzanie, testowanie, wdrożenie), a akcjami - przygotowanie artefaktów właściwych dla danej dyscypliny. Jeżeli w ramach jednej dyscypliny rozpoznano więcej artefaktów, przedstawić je na diagramie w postaci rozgałęzienia

Uwaga! "Rozgałęzienie" = przetwarzanie współbieżne, w przeciwieństwie do "decyzja" = rozgałęzienie warunkowe.

Wiktor Zychła



Rysunek 1: Diagram sekwencji do zadania 7