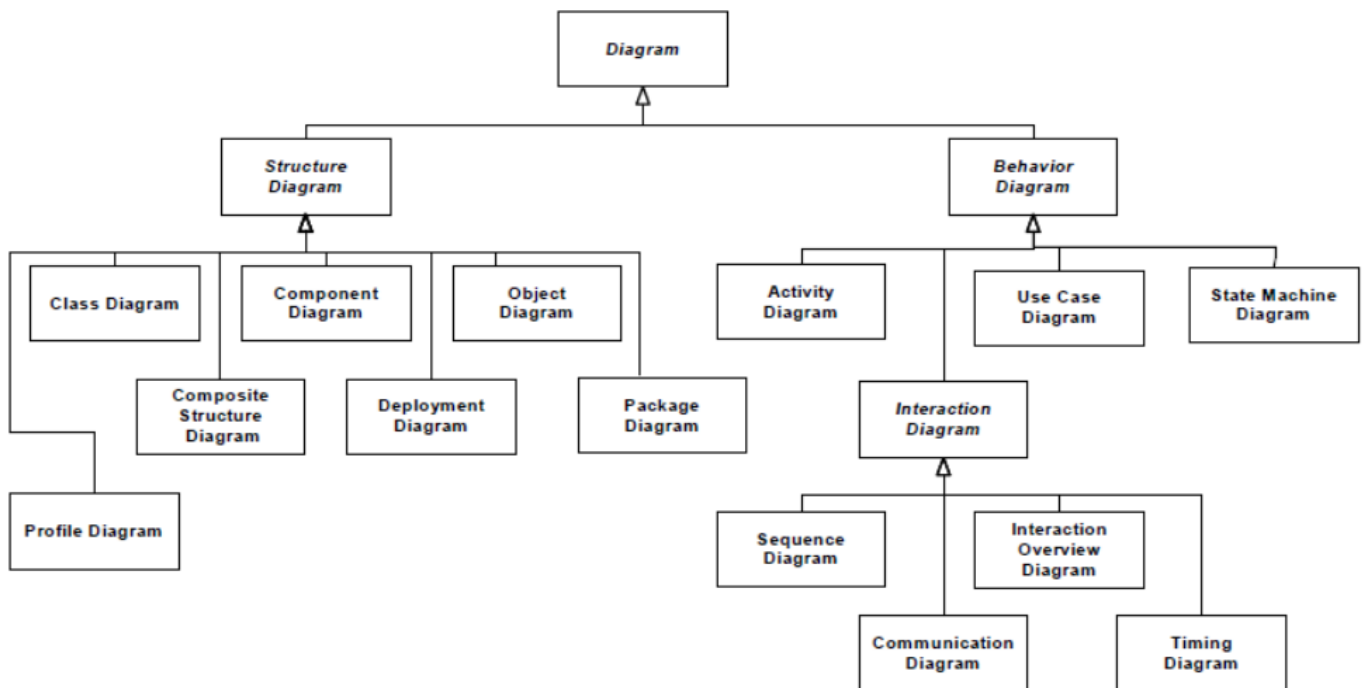


Zaawansowane Praktyki Programistyczne – lista zadań nr 4

opracował: dr inż. Jakub Długosz

Celem ćwiczenia jest stworzenie **diagramów UML** reprezentujących strukturę całego systemu na którym działa aplikacja do obsługi pracowników firmy FirmaX.

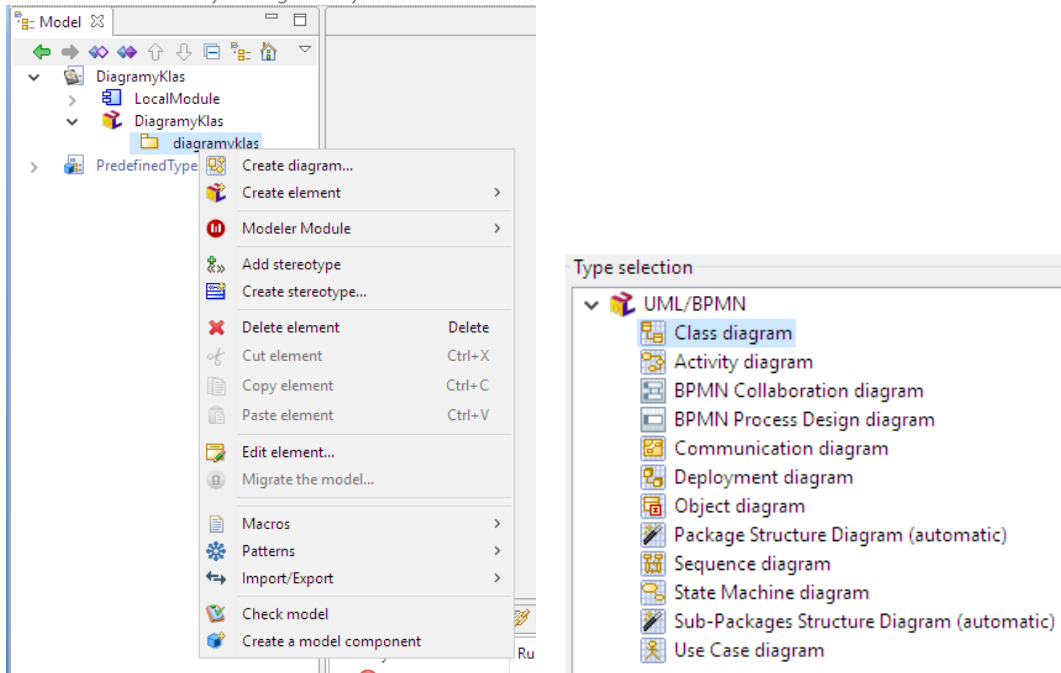
1. Do stworzenia diagramu należy wykorzystać język UML (Zunifikowany Język Modelowania) wg specyfikacji 2.5.1. Pełna specyfikacja dostępna jest na stronie: <https://www.omg.org/spec/UML/2.5.1/PDF> . Podstawowy podział diagramów (schematów) UML przedstawia poniższy rysunek (źródło: <https://www.omg.org/spec/UML/2.5.1/PDF>):



Na rysunku widać podstawowy podział na dwie zasadnicze gałęzie – diagramy strukturalne (statyczne, niezmiennie w czasie) i behawioralne (stany są zmienne w czasie).

2. Aby narysować diagramy UML można zainstalować darmowy program Modelio dostępny pod adresem <https://www.modelio.org/downloads/download-modelio.html> . Można wykorzystać też dowolny inny program pozwalający na tworzenie modeli zgodnych ze specyfikacją UML, można też wykonać rysunki odręcznie na kartce papieru.

3. W Modelio diagramy tworzymy klikając prawym klawiszem myszy na dany zasób w pakiecie w widoku modelu (Views > Model), następnie wybieramy polecenie „Create diagram...” i wybieramy z wyświetlonej listy rodzaj diagramu:



4. Zapoznaj się z diagramem wdrożenia (ang. Deployment Diagram) oraz pojęciami artefakt i węzeł.

Na diagramie przedstaw serwer aplikacyjny komunikujący się za pomocą protokołu HTTP z komputerem klienta. Serwer aplikacyjny łączy się za pomocą ODBC z serwerem bazy danych oraz za pomocą portu USB jest podłączony do drukarki.

Zilustruj fakt, że na komputerze klienta można korzystać z trzech różnych programów do komunikacji z serwerem aplikacji (możesz sam podać nazwy tych programów lub założyć, że są to przeglądarki: Internet Explorer, Google Chrome, Mozilla Firefox).

Uwzględnij fakt, że serwer aplikacyjny posiada plik konfiguracyjny „ustawienia.cfg” oraz korzysta z klas Osoba oraz Pracownik. Klasa Pracownik rozszerza klasę Osoba (klasa Pracownik dziedziczy z klasy Osoba). Klasy Osoba i Pracownik znajdują się we wspólnym pakiecie PakietOsobowy. Pełny kod klas na podstawie którego należy utworzyć specyfikację UML tych klas znajduje się w punkcie 5.

Uwzględnij fakt, że serwer bazy danych zawiera bazę danych FirmaX a w niej tabele SQL: osoby, pracownicy, wypłaty.

5. Poniżej znajduje się kod klas Osoba oraz Pracownik napisany w Javie. Na podstawie tego kodu należy utworzyć specyfikację UML tych klas.

```
public class Osoba {
    private String imie;
    private String nazwisko;
    private String telefon;

    public Osoba(String imie, String nazwisko, String telefon) {
        this.imie = imie;
        this.nazwisko = nazwisko;
        this.telefon = telefon;
    }

    public String getImie() {
        return imie;
    }
}
```

```

public void setImie(String imie) {
    this.imie = imie;
}

public String getNazwisko() {
    return nazwisko;
}

public void setNazwisko(String nazwisko) {
    this.nazwisko = nazwisko;
}

public String getTelefon() {
    return telefon;
}

public void setTelefon(String telefon) {
    this.telefon = telefon;
}
} //koniec klasy Osoba

public class Pracownik extends Osoba {
    private String stanowisko;

    public Pracownik(String imie, String nazwisko, String telefon, String stanowisko) {
        super(imie, nazwisko, telefon);
        this.stanowisko = stanowisko;
    }
} //koniec klasy Pracownik

```

6.

Wskazówka: W Modelio, mając zaznaczoną klasę, po kliknięciu na ikonę „Symbol”, możemy rozwinąć panel i zaznaczyć opcje „show attributes” i „show operations” aby utworzyć dodatkowe przegródki dla atrybutów i metod (operacji).

