

Opis przedmiotu:

Celem zajęć jest przedstawienie studentom praktyki projektowania obiektowego oprogramowania. Zajęcia w zwięzły sposób łączą podstawową wiedzę z zakresu baz danych i inżynierii oprogramowania w obszarze projektowania obiektowego w wybranych współczesnych realiach technologicznych.

Wymagania: przyda się znajomość paradygmatu obiektowego, pojęć obiektu, konstruktora, dziedziczenia, polimorfizmu i jego realizacji we współczesnych językach programowania.

Zajęcia są pomyślane jako zamiennik dla następujących kanonicznych zajęć:

- w części dotyczącej baz danych - Bazy danych
- w części dotyczącej projektowania obiektowego (wzorce projektowe) i inżynierii oprogramowania - Projektowanie obiektowe oprogramowania
- w części praktycznej dotyczącej wytwarzania aplikacji internetowych - specjalizowane kursy technologii wytwarzania aplikacji internetowych, m.in. kurs Projektowanie aplikacji ASP.NET, kurs Ruby on Rails, kurs WWW

Wykład

Część podstawowa - obejmuje materiał realizujący wymagania programowe:

1. podstawy projektowania obiektowego
 - a. analiza obiektowa
 - b. zbieranie wymagań
 - c. przypadki użycia
2. język UML
 - . diagramy klas, obiektów, zachowań i sekwencji
3. od modelu pojęciowego do modelu relacyjnego
 - . etap modelu pojęciowego
 - a. etap modelu obiektowego
 - b. etap modelu relacyjnego
4. podstawy praktyki baz danych
 - . tabele, kolumny, więzy
 - a. klucze podstawowe, klucze obce
 - b. widoki, procedury składowane
 - c. indeksy
 - d. język SQL - zapytania SELECT, INSERT, UPDATE, DELETE, podzapytania, klauzule WHERE, ORDER, TOP, JOIN, złączenia wewnętrzne i zewnętrzne
5. język Javascript - nowoczesny hybrydowy język funkcyjno-obiektowy

- . filozofia języka - Javascript jako praktyczna realizacja idei Javy - write once run everywhere
- a. historia, środowiska uruchomieniowe (Google V8, Microsoft Chakra)
- b. podstawowe elementy języka, zasięg zmiennych, hoisting
- c. system typów, dynamiczne/luźne typowanie
- d. obiekty, właściwości
- e. specyfika **this** - wiązanie kontekstowe
- f. listy/tablice
- g. enumeracja składowych obiektów vs enumeracja elementów tablic
- h. funkcje jako typy pierwszoklasowe (first class citizens)
- i. memoizacja
- j. redefinicje funkcji
- k. domknięcia
- l. tworzenie obiektów - funkcje konstruktorowe, prototypy, literały
- m. dziedziczenie/enkapsulacja
- n. IIFE (immediately-invoked function expressions)
- o. lambda wyrażenia
- p. klasy
- q. generatory
- r. callbacks vs promises vs async/await
- s. modularyzacja
- t. obiekty hostów - przeglądarka internetowa vs środowisko node.js
- u. debugowanie kodu w przeglądarce
- v. środowiska zintegrowane
- w. narzędzia jakościowe: JSLint/JSHint/Closure
- x. ekosystem/nadzbiorzy/transpilacja (Typescript/Coffeescript/ES6)
- y. Javascript jako język frontendowy i backendowy - kompilacja innych języków do Javascript (m.in. LLVM, emscripten, inne)
- 6. wzorce projektowe i wzorce architektury - przegląd
- 7. aplikacja internetowa w przeglądarce
 - . podstawy HTML5, elementy struktury dokumentu
 - a. podstawy CSS3, pozycjonowanie, wygląd, stylowanie responsywne
 - b. podstawy modelu Document Object Model (DOM)
 - c. model zdarzeniowy, pętla zdarzeń
 - d. obiekt canvas i podstawy grafiki bitmapowej
- 8. aplikacja internetowa na serwerze
 - . środowisko node.js, pakiety npm
 - a. podstawowe moduły, m.in. filesystem, http
 - b. framework Express - routing, wzorzec MVC, silnik widoków EJS
 - c. framework Socket.IO
 - d. autentykacja/autoryzacja
 - e. komunikacja z wybraną bazą danych
- 9. testy jednostkowe aplikacji Javascript - framework Mocha, technologia Selenium

Pracownia

Pracownia będzie polegała na uzupełnieniu materiału z wykładu szeregiem zadań o charakterze praktycznym. Przewiduje się listy zadań odpowiadające poszczególnym partiom materiału z wykładu.