Course description:

The aim of the course is to provide students with the practice of object-oriented software design. Classes in a concise way to combine basic knowledge of databases and software engineering in the area of object-oriented design in selected contemporary technological realities.

Requirements: useful knowledge of object-oriented paradigm, concepts object constructor, inheritance, and polymorphism its implementation in modern programming languages.

The classes are intended as a replacement for the following canonical classes:

- in the part concerning databases - Databases
- in the section on object-oriented design (design patterns), and software engineering - Designing Object-Oriented Software
- in the practical application relating to the preparation of web - specialized courses in technology of web applications, including ASP.NET Application Development course, the course Ruby on Rails Web course

Lecture

Basic part - obiejmuje material implementing the program requirements:

1. basics of object-oriented design
   and. object-oriented analysis
   b. requirements gathering
   c. use cases
2. UML
   . class diagrams, object, behavior and sequence
3. from the conceptual model to the relational model
   . stage conceptual model
   and. Stage object model
   b. stage of the relational model
4. basic practices database
   . tables, columns, constraints
   and. primary keys, foreign keys
   b. views, stored procedures
   c. indexes
   d. SQL - the query SELECT, INSERT, UPDATE, DELETE, subqueries, WHERE clauses, ORDER, TOP, JOIN, join internal and external
5. Javascript language - a hybrid modern object-oriented language a functionality

. philosophy of language - Javascript as a practical realization of the idea of Java - write once run everywhere

and. history, runtime environments (Google V8, Microsoft's Chakra)
b. the basic elements of language, range of variables, hoisting
c. type system, dynamic / loose typing
d. objects, properties
e. specifics this - context bond
f. lists / arrays
g. enumeration object members vs enumeration array elements
h. functions as types pierwszoklasowe (first class citizens)
and. memoizacja
j. redefinitions of functions
k. closures
l. create objects - functions constructors, prototypes, literals
m. Inheritance / encapsulation
n. Iife (Immediately-invoked function expressions)
about. lambda expressions
p. class
q. generators
r. callbacks promises vs vs async / await
p. modularization
t. Host objects - web browser environment vs node.js
at. debugging code in the browser
v. integrated environment
in. quality tools: JSLint / JSHint / Closure
x. ecosystem / supersets / transpilacja (Typescript / CoffeeScript / ES6)
s. Javascript as a language and frontendowy backendowy - other languages compile to JavaScript (including LLVM, emscripten, etc.)

6. design patterns and architectural patterns - review
7. web application in the browser

. basic HTML5 document structure elements
and. CSS3 base, positioning, appearance, stylish responsive
b. base Document Object Model (DOM)
c. The event model, the event loop
d. Canvas object and the base bitmap graphics

8. web application server

. Environment node.js, packages asl
and. basic modules, including filesystem, http
b. Express framework - routing pattern MVC view engine EJS
c. framework Socket.IO
d. authentication / authorization
e. communication with your database

9. Javascript unit testing applications - Mocha framework, Selenium technology

Lab

Workshop will be to complement the lecture material from a number of practical tasks. It provides a list of tasks corresponding to different parties of the lecture material.