

# IT systems design - opis przedmiotu

| Informacje ogólne   |                                                                   |
|---------------------|-------------------------------------------------------------------|
| Nazwa przedmiotu    | IT systems design                                                 |
| Kod przedmiotu      | 11.3-WE-INFP-ITSD- Er                                             |
| Wydział             | <a href="#">Wydział Informatyki, Elektrotechniki i Automatyki</a> |
| Kierunek            | Informatyka                                                       |
| Profil              | ogólnoakademicki                                                  |
| Rodzaj studiów      | Program Erasmus pierwszego stopnia                                |
| Semestr rozpoczęcia | semestr zimowy 2019/2020                                          |

| Informacje o przedmiocie        |                                                                                                             |
|---------------------------------|-------------------------------------------------------------------------------------------------------------|
| Semestr                         | 5                                                                                                           |
| Liczba punktów ECTS do zdobycia | 6                                                                                                           |
| Typ przedmiotu                  | obowiązkowy                                                                                                 |
| Język nauczania                 | angielski                                                                                                   |
| Sylabus opracował               | <ul style="list-style-type: none"><li>dr inż. Tomasz Gratkowski</li><li>dr inż. Michał Doligalski</li></ul> |

| Formy zajęć  |                                            |                                           |                                               |                                              |                     |
|--------------|--------------------------------------------|-------------------------------------------|-----------------------------------------------|----------------------------------------------|---------------------|
| Forma zajęć  | Liczba godzin w semestrze<br>(stacjonarne) | Liczba godzin w tygodniu<br>(stacjonarne) | Liczba godzin w semestrze<br>(niestacjonarne) | Liczba godzin w tygodniu<br>(niestacjonarne) | Forma zaliczenia    |
| Wykład       | 30                                         | 2                                         | -                                             | -                                            | Egzamin             |
| Laboratorium | 30                                         | 2                                         | -                                             | -                                            | Zaliczenie na ocenę |

## Cel przedmiotu

- To introduce students with the basics of software engineering and software modeling methods.
- Formation of the students understanding of the principles of object-oriented programming.
- To familiarize students with the principles of compiler design.

## Wymagania wstępne

Principles of programming, Software engineering

## Zakres tematyczny

Elements of software engineering.

Software development. Software crisis and ways to tackle them. Conceptual modeling. The role of modeling in software development. Historical background of modern modeling techniques. Object-oriented design methods and UML notation. Structured and object-oriented methodology.

Business process modeling in BPMN. Create a software model based on BPMN model. Analysis and requirements modeling. Domain analysis and modeling. Architecture design. Software life cycle. System Design and System Analysis. The basic object-oriented concepts and relationships between objects. Modeling links between objects. Messages and procedure calls. Classes, inheritance, generalization / specialization, polymorphism, interfaces.

Unified Modelling Language UML. The genesis of the uprising. Definition and objectives of the creation of UML. The scope of UML. UML diagrams. Characteristic diagrams.

## Metody kształcenia

Lecture: conventional lecture

Laboratory: laboratory exercises, group work

## Efekty uczenia się i metody weryfikacji osiągnięcia efektów uczenia się

| Opis efektu                                                                         | Symbole efektów | Metody weryfikacji                                                            | Forma zajęć                                                  |
|-------------------------------------------------------------------------------------|-----------------|-------------------------------------------------------------------------------|--------------------------------------------------------------|
| Student is able to prepare design documentation for IT system.                      |                 | <ul style="list-style-type: none"><li>bieżąca kontrola na zajęciach</li></ul> | <ul style="list-style-type: none"><li>Laboratorium</li></ul> |
| Can develop customer requirements for the IT system.                                |                 | <ul style="list-style-type: none"><li>bieżąca kontrola na zajęciach</li></ul> | <ul style="list-style-type: none"><li>Laboratorium</li></ul> |
| Can customize the methodology of project management to the design of the IT system. |                 | <ul style="list-style-type: none"><li>sprawdzian</li><li>test</li></ul>       | <ul style="list-style-type: none"><li>Wykład</li></ul>       |
| Student can modeling any IT system using UML.                                       |                 | <ul style="list-style-type: none"><li>bieżąca kontrola na zajęciach</li></ul> | <ul style="list-style-type: none"><li>Laboratorium</li></ul> |

## Warunki zaliczenia

Lecture - obtaining a positive grade in written exam.

Laboratory - the main condition to get a pass are sufficient marks for all exercises and tests conducted during the semester.

Calculation of the final grade: = lecture 50% + laboratory 50%.

## Literatura podstawowa

- Alan Dennis, Barbara Haley Wixom, David Tegarden: Systems Analysis and Design: An Object-Oriented Approach with UML, Wiley 2015
- Brett D. McLaughlin, Gary Pollice, Dave West: Head First Object-Oriented Analysis and Design, imusti 2006
- UML Distilled: A Brief Guide to the Standard Object Modeling Language (3rd Edition)
- By Martin Fowler: UML Distilled: A Brief Guide to the Standard Object Modeling Language (3rd Edition), Addison-Wesley Professional; 2003
- Stephen Schach: Object-Oriented and Classical Software Engineering (Irwin Computer Science), McGraw-Hill Education; 8 edition (July 19, 2010)

## Literatura uzupełniająca

## Uwagi

Zmodyfikowane przez prof. dr hab. inż. Andrzej Obuchowicz (ostatnia modyfikacja: 27-10-2019 10:39)

Wygenerowano automatycznie z systemu SylabUZ