

Object-oriented Programming 1 - course description

General information	
Course name	Object-oriented Programming 1
Course ID	11.3-WK-IIeP-PO1-L-S14_pNadGen864Q8
Faculty	Faculty of Mathematics, Computer Science and Econometrics
Field of study	Informatics and Econometrics
Education profile	academic
Level of studies	First-cycle studies leading to Bachelor's degree
Beginning semester	winter term 2018/2019

Course information	
Semester	5
ECTS credits to win	6
Course type	optional
Teaching language	polish
Author of syllabus	dr inż. Janusz Jabłoński

Classes forms					
The class form	Hours per semester (full-time)	Hours per week (full-time)	Hours per semester (part-time)	Hours per week (part-time)	Form of assignment
Laboratory	30	2	-	-	Credit with grade
Lecture	30	2	-	-	Exam

Aim of the course

Opanowanie umiejętności pisania prostych programów w języku Java.

Prerequisites

Programowanie komputerów 2.

Scope

Wykład

1. Historia Javy.
2. Podstawy: typy danych, zmienne, instrukcje przypisania, operatory.
3. Łańcuchy i ich metody. Instrukcje sterujące.
4. Tablice i operacje na tablicach.
5. Tworzenie własnych klas: konstruktory, metody prywatne, statyczne, publiczne, fabryczne oraz metoda main, pola.
6. Tworzenie i niszczenie obiektów.
7. Pakiety.
8. Komentarze i generowanie dokumentacji.
9. Klasy i metody finalne. Interfejsy a klasy abstrakcyjne.
10. Dziedziczenie. AWT a Swing. Obsługa zdarzeń.

Laboratorium

1. Zapoznanie studentów z środowiskami programistycznymi.
2. Pisanie i uruchamianie: prostych klas z instrukcjami przypisania, zawierające instrukcje warunkowe i iteracyjne.
3. Przetwarzanie tablic jednowymiarowych i dwuwymiarowych.
4. Operacje na łańcuchach znaków.
5. Debugowanie: śledzenie krok po kroku instrukcji programu i podglądanie wartości zmiennych.
6. Stosowanie interfejsów.
7. Tworzenie klas, które dziedziczą po istniejących już klasach.

Teaching methods

Wykład: wykład konwersatoryjny.

Laboratorium: ćwiczenia laboratoryjne w pracowni komputerowej – pisanie i uruchamianie samodzielnie napisanych programów oraz analiza tych programów.

Learning outcomes and methods of theirs verification

Outcome description	Outcome symbols	Methods of verification	The class form
---------------------	-----------------	-------------------------	----------------

Outcome description	Outcome symbols	Methods of verification	The class form
Student potrafi definiować proste klasy w Javie i umie korzystać z istniejących	• K_U26	• a discussion • an evaluation test • an exam - oral, descriptive, test and other • an ongoing monitoring during classes • kartkówka	• Lecture • Laboratory
Student potrafi uruchamiać i debugować programy napisane w Javie	• K_U26	• an ongoing monitoring during classes	• Laboratory
Student potrafi tworzyć diagramy klas	• K_U25	• a discussion • an evaluation test • an exam - oral, descriptive, test and other • an ongoing monitoring during classes • kartkówka	• Lecture • Laboratory
Student dobrze zna podstawowe mechanizmy programowania zorientowanego obiektowo w Javie.	• K_W10	• a discussion • an evaluation test • an exam - oral, descriptive, test and other • an ongoing monitoring during classes • kartkówka	• Lecture • Laboratory

Assignment conditions

Wykład: egzamin złożony z dwóch części pisemnej i ustnej, warunkiem przystąpienia do części ustnej jest uzyskanie 30% punktów z części pisemnej, uzyskanie 50% punktów z części pisemnej gwarantuje uzyskanie pozytywnej oceny.

Laboratorium: warunkiem zaliczenia jest uzyskanie ponad 50% punktów z przeprowadzonych w semestrze kartkówek lub kolokwium obejmującego cały przerabiany materiał.

Ocena końcowa **przedmiotu jest średnią arytmetyczną ocen z egzaminu i laboratorium. Jednakże warunkiem koniecznym uzyskania pozytywnej oceny końcowej jest uzyskanie pozytywnej oceny**

Recommended reading

1. C. Horstmann, G. Cornell, Java 2. Podstawy, Helion, 2003.
2. K. Sierra, B. Bates, Head First Java. Helion, 2004.
3. M. Lis, Java. Ćwiczenia praktyczne. Helion, 2006.

Further reading

1. C. Horstmann, G. Cornell, Java 2. Techniki zaawansowane, Helion, 2003.
2. B. Eckel, Thinking in Java. Helion, 2001.

Notes

Modified by dr Robert Dylewski, prof. UZ (last modification: 04-05-2018 19:24)

Generated automatically from SyllabUZ computer system