

Nazwa przedmiotu	Advanced Object Programming in Java														
Nazwa w j zyku angielskim	Advanced Object Programming in Java														
J zyk prowadzenia zaj	angielski														
Poziom studiów	studia drugiego stopnia														
Profil studiów	A, ogólnoakademicki														
Jednostka prowadz ca	Instytut Informatyki Stosowanej														
Kierownik i realizatorzy	Bieniecki Wojciech, dr in . Grabowski Szymon, dr hab.														
Formy zaj i liczba godzin w semestrze	<table border="1"> <thead> <tr> <th>Wyk.</th> <th>w.</th> <th>Lab.</th> <th>Proj.</th> <th>Sem.</th> <th>Inne</th> <th>Suma godzin w semestrze</th> </tr> </thead> <tbody> <tr> <td>15</td> <td>0</td> <td>15</td> <td>0</td> <td>0</td> <td>0</td> <td>30</td> </tr> </tbody> </table>	Wyk.	w.	Lab.	Proj.	Sem.	Inne	Suma godzin w semestrze	15	0	15	0	0	0	30
Wyk.	w.	Lab.	Proj.	Sem.	Inne	Suma godzin w semestrze									
15	0	15	0	0	0	30									
Cele przedmiotu	Zaznajomienie studentów z zaawansowanymi technikami programowania w Javie: - wzorce projektowe, - programowanie funkcyjne, - programowanie rozproszone.														
po zmianie	1. Zaznajomienie studentów z zaawansowanymi technikami programowania w Javie takimi jak - wzorce projektowe, 2. Zaznajomienie studentów z zaawansowanymi technikami programowania w Javie takimi jak - programowanie funkcyjne, 3. Zaznajomienie studentów z zaawansowanymi technikami programowania w Javie takimi jak - programowanie rozproszone.														
Efekty kształcenia	Po zako czeniu kursu student b dzie w stanie: 1. Wyja nia subtelno ci programowania obiektowego. 2. Stosowa odpowiednie wzorce projektowe. 2. Budowa aplikacje rozproszone z wykorzystaniem odpowiednich technologii Javy. 3. Projektowa i implementowa zło one aplikacje z zastosowaniem programowania funkcjonalnego. 5. Rozwi zywa praktyczne zadania z u yciem zaawansowanej biblioteki kolekcji Google Guava.														
po zmianie	1. Student wyja nia subtelno ci programowania obiektowego. 2. Student stosuje odpowiednie wzorce projektowe. 3. Student buduje aplikacje rozproszone z wykorzystaniem odpowiednich technologii Javy. 4. Student projektuje i implementuje zło one aplikacje z zastosowaniem programowania funkcjonalnego.														
Metody weryfikacji efektów kształcenia	Efekt 1 i 2 - kolokwium wykładowe. Efekt 3, 4 i 5 - samodzielne zadania laboratoryjne i domowe.														
po zmianie	Efekt 1 i 2 - kolokwium wykładowe. Efekt 3, 4 - samodzielne zadania laboratoryjne i domowe.														
Wymagania wst pne	Podstawowa umiej tno programowania obiektowego w Javie, znajomo podstaw in ynierii oprogramowania, w tym modelowania i analizy obiektowej														
Organizacja przedmiotu i tre ci kształcenia	WYKŁAD: 1. Zaawansowane mechanizmy obiektowe w j zyku Java 2. Wzorce projektowe														

3. Aplikacje rozproszone
4. Programowanie funkcyjne
5. Biblioteki rozszerzaj ce j zyk Java

LABORATORIUM

Laboratorium składa si z podstawowych wicze laboratoryjnych i trudniejszych zada domowych realizuj cych tre ci wykładu.

po zmianie

WYKŁAD

- paradygmaty programowania zaimplementowane w Javie
- mechanizm wyj tków jako przykład programowania zdarzeniowego
- Struktury danych - kolekcje, tablice, listy, zbiory, słowniki
- nowe elementy Java8 - strumienie, elementy programowania funkcyjnego
- Programowanie generyczne
- Refleksja
- Adnotacje
- Aplikacje sieciowe i bazodanowe.

WICZENIA LABORATORYJNE

1. aplikacje w Javie z u yciem pakietów, klas, własnych wyj tków i testów jednostkowych. Tworzenie dokumentacji JavaDoc.
2. Implementacja zada algorytmicznych z wykorzystaniem kolekcji oraz strumieni Java 8.
3. Strumienie plikowe: formaty tekstowe, binarne, obiektowe i skompresowane.
4. Klasy dynamiczne, refleksja i adnotacje czasu wykonania.
5. Aplikacja sieciowa wielow tkowa.
6. Aplikacja bazodanowa z mechanizmem ORM.

Formy zaliczenia - sprawdzenie osi gni efektów kształcenia

Kolokwium wykładowe (50%), zadania laboratoryjne i domowe (50%).

Literatura podstawowa

C. S. Horstmann, G. Cornell: Core Java, Vol. 2: Advanced Feature, Prentice Hall 2008.
E. Gamma et al.: Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley Professional, 1994.
E. Freeman et al.: Head First Design Patterns, O&aposp;Reilly Media, 2004.

po zmianie

1. C. S. Horstmann, G. Cornell: Core Java, Vol. 2: Advanced Feature, Prentice Hall 2008.
2. E. Gamma et al.: Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley Professional, 1994.
3. E. Freeman et al.: Head First Design Patterns, O&aposp;Reilly Media, 2004.

Literatura uzupełniaj ca

<http://code.google.com/p/lambdaj/>
<http://code.google.com/p/guava-libraries/>

po zmianie

1. <http://code.google.com/p/lambdaj/>
2. <http://code.google.com/p/guava-libraries/>

Przeci tne obci enie studenta prac własn - ze zdefiniowaniem form pracy własnej

Suma godzin wszystkich form zaj	30
Udział w konsultacjach	5
Udział w pisemnych i/lub praktycznych formach weryfikacji	1
Przygotowanie do kolokwium wykładowego	12
Przygotowanie do laboratorium	15
Przygotowanie prac domowych	15
Suma godzin:	78
Suma godzin powinna mie ci si w zakresie:	75..90

Uwagi

*Uwagi własne
publikowane*

po zmianie

brak

Data aktualizacji

2013-05-15 13:10:43

Course name **Advanced Object Programming in Java**

Course name in Polish **Advanced Object Programming in Java**

Language of instruction English

Level of studies second-cycle programme

Type of studies nie zdefiniowano

Unit running the programme Instytut Informatyki Stosowanej

Course coordinator and academic teachers

Bieniecki Wojciech, dr in .

Grabowski Szymon, dr hab.

Form of classes and number of teaching hour per semester

Lec.	Tut.	Lab.	Proj.	Sem.	Other	Total number of teaching hour per semester
15	0	15	0	0	0	30

Goals The course aims to acquaint students with advanced programming techniques in Java:
 - design patterns,
 - elements of functional programming,
 - elements of distributed programming.

po zmianie 1. To acquaint students with advanced programming techniques in Java just as - design patterns,
 2. To acquaint students with advanced programming techniques in Java just as - elements of functional programming,
 3. To acquaint students with advanced programming techniques in Java just as - elements of distributed programming.

Learning outcomes A student at the end of the course will be able to:
 1. Explain subtleties of object-oriented programming.
 2. Employ design patterns.
 3. Build distributed applications using appropriate Java technologies.
 4. Design and implement complex programs using functional paradigms.
 5. Solve practical problems using Google Guava advanced collection library.

after changes 1. Student explains subtleties of object-oriented programming.
 2. Student employs design patterns.
 3. Student builds distributed applications using appropriate Java technologies.
 4. Student designs and implements complex programs using functional paradigms.

Learning outcomes verification methods Effects 1 and 2 - lecture test.
 Effects 3, 4, and 5 - lab/home assignments.

after changes Effects 1 and 2 - lecture test.
 Effects 3, 4 - lab/home assignments.

Prerequisites Basic capabilities of Java object-oriented programming, knowlegde of fundamentals of software engineering, including modelling and object-oriented analysis

Course organisation and content LECTURE:
 1. Advanced OOP mechanisms in Java.
 2. Design patterns.
 3. Distributed applications.
 4. Functional programming.
 5. Advanced Java libraries.

LABORATORY:
Basic lab assignments and (harder) home problems, covering the lecture content.

after changes

LECTURE
- programming paradigms existing in Java
- exceptions such as an event-programming example
- data structures - collections, tables, lists, sets, dictionaries
- new elements of Java 8 - streams, elements of functional programming
- generic programming
- reflection
- annotations
- network and database applications.

LABORATORY:
1. applications in Java using packages, classes, own exceptions and unit tests. Documentation with JavaDoc.
2. Implementation of algorithmic tasks with collections and Java 8 streams.
3. File Streams: text, binary, object and compressed formats.
4. Dynamic classes, reflection and runtime annotations.
5. multi-threaded network application.
6. database applications with ORM mechanism.

Form of assessment

Lecture test (50%), lab and home assignments (50%).

Basic reference materials

C. S. Horstmann, G. Cornell: Core Java, Vol. 2: Advanced Feature, Prentice Hall 2008.
E. Gamma et al.: Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley Professional, 1994.
E. Freeman et al.: Head First Design Patterns, O'Reilly Media, 2004.

after changes

1. C. S. Horstmann, G. Cornell: Core Java, Vol. 2: Advanced Feature, Prentice Hall 2008.
2. E. Gamma et al.: Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley Professional, 1994.
3. E. Freeman et al.: Head First Design Patterns, O'Reilly Media, 2004.

Other reference materials

<http://code.google.com/p/lambdaj/>
<http://code.google.com/p/guava-libraries/>

after changes

1. <http://code.google.com/p/lambdaj/>
2. <http://code.google.com/p/guava-libraries/>

Average student work-load outside classroom

Total hours of different forms of classes	30
Participation in consultations	5
Participation in written and/or practical forms of assesment	1
Lecture test preparation	12
Lab preparation	15
Home assignments	15
Total hours:	78
Total hours should be in the range:	75..90

Published comments

after changes

brak

Update date

2013-05-15 13:10:43