

KARTA PRZEDMIOTU**I. Dane podstawowe**

Nazwa przedmiotu	Laboratorium programowania – aplikacje w różnych środowiskach programistycznych
Nazwa przedmiotu w języku angielskim	Laboratory of programming: applications in basic programming environments
Kierunek studiów	Informatyka w j. angielskim
Poziom studiów (I, II, jednolite magisterskie)	I stopnia
Forma studiów (stacjonarne, niestacjonarne)	Stacjonarne
Dyscyplina	Informatyka, informatyka techniczna i telekomunikacja
Język wykładowy	Angielski

Koordinator przedmiotu/osoba odpowiedzialna	dr Michał Dolecki
---	-------------------

Forma zajęć (<i>katalog zamknięty ze słownika</i>)	Liczba godzin	Semestr	Punkty ECTS
wykład			3+3
konwersatorium			
ćwiczenia			
laboratorium	30+30	V+VI	
warsztaty			
seminarium			
proseminarium			
lektorat			
praktyki			
zajęcia terenowe			
pracownia dyplomowa			
translatorium			
wizyta studyjna			

Wymagania wstępne	Basic knowledge in Object-oriented programming and Data Bases. Basic knowledge in mathematics, physics and computer science.
-------------------	---

II. Cele kształcenia dla przedmiotu

Developing programming skills in object-oriented language.
Getting to know the characteristics and features of Java language.

III. Efekty uczenia się dla przedmiotu wraz z odniesieniem do efektów kierunkowych

Symbol	Opis efektu przedmiotowego	Odniesienie do efektu kierunkowego
WIEDZA		
W_01	The student lists the advantages of the correct and responsible programming and testing for the good functioning of the team	K_W09
UMIEJĘTNOŚCI		
U_01	The student can implement different algorithms and use them for solving problems from various areas of life	K_U08
U_02	The student can work with technical documentation and Internet to find and understand the solution for particular problem	K_U02
KOMPETENCJE SPOŁECZNE		
K_01	The student can analyze the problem, decompose it into packages and classes with clear responsibilities and determinate the order of its implementation	K_K02

IV. Opis przedmiotu/ treści programowe

<p>Features of object-oriented programming Unit tests and test driven development Tests to check the exceptions Creating an appropriate hierarchy of classes for given problem Class diagrams The use of abstract classes and interfaces The use of static fields and methods Increasing the code reusability Three-tier application model Data storage layer modeling, creating business logic tier and presentation design Creating a GUI in Java</p>

V. Metody realizacji i weryfikacji efektów uczenia się

Symbol efektu	Metody dydaktyczne <i>(lista wyboru)</i>	Metody weryfikacji <i>(lista wyboru)</i>	Sposoby dokumentacji <i>(lista wyboru)</i>
WIEDZA			
W_01	Guided practice	Preparation / implementation of the project, Written test	Project rating card, Evaluated written test
UMIEJĘTNOŚCI			
U_01	Practical classes	Preparation / implementation of the project, Written test	Project rating card, Evaluated written test
U_02	Practical classes	Preparation / implementation of the project, Written test	Project rating card, Evaluated written test

KOMPETENCJE SPOŁECZNE			
K_01	Discussion, PBL (Problem-Based Learning)	Preparation / implementation of the project, Written test	Project rating card, Evaluated written test

VI. Kryteria oceny, wagi...

Colloquium - 40%

Final project - 40%

Activity - 20%

0% - 49% - unsatisfactory (2.0)

50% - 59% - satisfactory (3.0)

60% - 69% - satisfactory plus (3.5)

70% - 79% - good (4.0)

80% - 89% - good plus (4.5)

90% - 100% - very good (5.0)

VII. Obciążenie pracą studenta

Forma aktywności studenta	Liczba godzin
Liczba godzin kontaktowych z nauczycielem	50+50
Liczba godzin indywidualnej pracy studenta	30+30

VIII. Literatura

Literatura podstawowa
B. Eckel, Thinking in Java, Prentice Hall, 1998+
C. Horstmann, G. Cornell, Core Java 2, Volume I: Fundamentals, Prentice Hall, 1999+
C. Horstmann, Core Java, Volume II: Advanced Features, Prentice Hall, 1999+
https://docs.oracle.com/javase/tutorial/
Literatura uzupełniająca