



KARTA MODUŁU / KARTA PRZEDMIOTU

Kod modułu	
Nazwa modułu	Programowanie imperatywne, obiektowe i deklaratywne
Nazwa modułu w języku angielskim	Imperative, object-oriented and declarative programming
Obowiązuje od roku akademickiego	2012/13

A. USYTUOWANIE MODUŁU W SYSTEMIE STUDIÓW

Kierunek studiów	Informatyka
Poziom kształcenia	II stopień (I stopień / II stopień)
Profil studiów	Ogólnoakademicki (ogólno akademicki / praktyczny)
Forma i tryb prowadzenia studiów	Niestacjonarne (stacjonarne / niestacjonarne)
Specjalność	Systemy informacyjne
Jednostka prowadząca moduł	Katedra Zastosowań Informatyki
Koordynator modułu	dr inż. Andrzej Kułakowski
Zatwierdził:	

B. OGÓLNA CHARAKTERYSTYKA PRZEDMIOTU

Przynależność do grupy/bloku przedmiotów	Kierunkowy (podstawowy / kierunkowy / inny HES)
Status modułu	Nieobowiązkowy (obowiązkowy / nieobowiązkowy)
Język prowadzenia zajęć	Polski
Usytuowanie modułu w planie studiów - semestr	3 lub 4
Usytuowanie realizacji przedmiotu w roku akademickim	Semestr zimowy lub letni (semestr zimowy / letni)
Wymagania wstępne	Podstawy programowania 2 Programowanie w języku C 2 Programowanie obiektowe (Java)
Egzamin	nie (tak / nie)
Liczba punktów ECTS	4

Forma prowadzenia zajęć	wykład	ćwiczenia	laboratorium	projekt	inne
w semestrze	18			18	



C. EFEKTY KSZTAŁCENIA I METODY SPRAWDZANIA EFEKTÓW KSZTAŁCENIA

Cel modułu	Poznanie różnych paradygmatów programowania oraz wybranych języków programowania dla paradygmatów: imperatywnego, obiektowego, funkcyjnego i logicznego. Uświadomienie rozmaitych aspektów stosowania i możliwości wykorzystania różnych paradygmatów programowania w określonych zastosowaniach (3-4 linijki)
-------------------	--

Symbol efektu	Efekty kształcenia	Forma prowadzenia zajęć (w/ć/l/p/inne)	odniesienie do efektów kierunkowych	odniesienie do efektów obszarowych
W_01	Ma podstawową znajomość różnych paradygmatów programowania	W	K_W03, K_W11	T2A_W03, T2A_W04, T2A_W05,
W_02	Ma podstawową wiedzę na temat różnych języków programowania	W	K_W03, K_W11	T2A_W03, T2A_W04, T2A_W05,
U_01	Potrafi zauważyć potrzebę stosowania różnych paradygmatów programowania	P	K_U01, K_U08	T2A_U01, T2A_U09, T2A_U18,
U_02	Potrafi wykorzystać nabytą wiedzę programistyczną do implementacji prostych programów dla różnych paradygmatów.	P	K_U08, K_U14	T2A_U09, T2A_U15, T2A_U16, T2A_U18,
K_01	Umiejętność projektowania programów w zespołach programistycznych.	P	K_K02	T2A_K03

Treści kształcenia:

1. Treści kształcenia w zakresie wykładu

Nr wykładu	Treści kształcenia	Odniesienie do efektów kształcenia dla modułu
1	Wprowadzenie. Paradygmaty programowania. Sposoby opisu języków programowania. Składnia, semantyka, kompilacja i interpretacja.	W_01, W_02
2	Programowanie imperatywne. Zmienne, wiązania, zgodność typów, zakres widoczności, semantyka zmiennych. Typy danych, implementacja typów danych, abstrakcyjne typy danych.	W_01, W_02
3	Wyrażenia, instrukcje, przepływ sterowania, podprogramy. Programowanie obiektowe, Klasy i polimorfizm, problemy implementacji. Języki obiektowe.	W_01, W_02
4	Programowanie deklaratywne - programowanie w logice. Prolog: wprowadzenie, programowanie, zmienne i ich ukonkretnianie, interpreter.	W_01, W_02
5	Prolog - elementy języka programowania. Zarządzanie danymi, rekurencja i struktury, listy. Przykłady programów. Środowisko programowania.	W_01, W_02
6	Elementy systemu ekspertowego. Przykład systemu ekspertowego w Prologu.	W_01, W_02
7	Programowanie funkcyjne. Wprowadzenie. Funkcje jako model programowania. Rachunek lambda. Przegląd języków funkcyjnych.	W_01, W_02
8	Wprowadzenie do przykładów programowania funkcyjnego. Typy danych. Rekurencja i listy. Wejście i wyjście. Lisp, Scheme, Haskell.	W_01, W_02
9	Podsumowanie. Inne paradygmaty programowania. Kolokwium.	W_01, W_02



2. Treści kształcenia w zakresie ćwiczeń

Nr zajęć ćwicz.	Treści kształcenia	Odniesienie do efektów kształcenia dla modułu

3. Treści kształcenia w zakresie zadań laboratoryjnych

Nr zajęć lab.	Treści kształcenia	Odniesienie do efektów kształcenia dla modułu

4. Charakterystyka zadań projektowych

Tematyka zagadnień projektowych obejmuje stworzenie trzech aplikacji wraz z dokumentacją na wybrany temat dotyczący jednego z zagadnień związanych z programowaniem imperatywnym, obiektowym, logicznym lub funkcyjnym.

- 1) Program z bazą danych w języku obiektowym.
- 2) Prosty system ekspertowy w prologu.
- 3) Program w wybranym języku funkcyjnym.
(U_01, U_02, K_01).

5. Charakterystyka zadań w ramach innych typów zajęć dydaktycznych

Metody sprawdzania efektów kształcenia

Symbol efektu	Metody sprawdzania efektów kształcenia (sposób sprawdzenia, w tym dla umiejętności – odwołanie do konkretnych zadań projektowych, laboratoryjnych, itp.)
W_01	Kolokwium
W_02	Kolokwium
U_01	na podstawie wyników realizacji projektu i odpowiedzi ustnej
U_02	na podstawie wyników realizacji projektu i odpowiedzi ustnej
K_01	Zaliczenie na podstawie wyników realizacji zagadnień oraz odpowiedzi ustnej



D. NAKŁAD PRACY STUDENTA

Bilans punktów ECTS		
	Rodzaj aktywności	obciążenie studenta
1	Udział w wykładach	18
2	Udział w ćwiczeniach	
3	Udział w laboratoriach	
4	Udział w konsultacjach (2-3 razy w semestrze)	4
5	Udział w zajęciach projektowych	18
6	Konsultacje projektowe	
7	Udział w egzaminie	
8		
9	Liczba godzin realizowanych przy bezpośrednim udziale nauczyciela akademickiego	40 (suma)
10	Liczba punktów ECTS, którą student uzyskuje na zajęciach wymagających bezpośredniego udziału nauczyciela akademickiego <i>(1 punkt ECTS=25-30 godzin obciążenia studenta)</i>	1,54
11	Samodzielne studiowanie tematyki wykładów	35
12	Samodzielne przygotowanie się do ćwiczeń	
13	Samodzielne przygotowanie się do kolokwium	10
14	Samodzielne przygotowanie się do laboratoriów	
15	Wykonanie sprawozdań	
15	Przygotowanie do kolokwium końcowego z laboratorium	
17	Wykonanie projektu lub dokumentacji	45
18	Przygotowanie do egzaminu	
19		
20	Liczba godzin samodzielnej pracy studenta	90 (suma)
21	Liczba punktów ECTS, którą student uzyskuje w ramach samodzielnej pracy <i>(1 punkt ECTS=25-30 godzin obciążenia studenta)</i>	3,46
22	Sumaryczne obciążenie pracą studenta	130
23	Punkty ECTS za moduł <i>1 punkt ECTS=25-30 godzin obciążenia studenta</i>	5 (4)
24	Nakład pracy związany z zajęciami o charakterze praktycznym <i>Suma godzin związanych z zajęciami praktycznymi</i>	77
25	Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym <i>1 punkt ECTS=25-30 godzin obciążenia studenta</i>	2,96



E. LITERATURA

Wykaz literatury	1. Van Roy P., Haridi S.: „Programowanie Koncepty, techniki i modele”, Helion, 2005. 2. Przegląd języków i paradygmatów programowania Jarosław Bylina, Beata Bylina, Instytut Informatyki UMCS, 2011 3. Meyer B.: „Programowanie zorientowane obiektowo”, Helion, 2005. 4. Kernigham B. W., Ritchie D. M.: „Język ANSI C”, WNT, Warszawa, 2004. 5. Stroustrup B.: „Język C++”, WNT, Warszawa, 2004. 6. Eckel B.: „Thinking in Java”, Helion, 2006. 7. Horstmann C. S., Cornell G.: „Java Podstawy”, Helion, 2008. 8. Real World Haskell, Code You Can Believe In, Bryan O'Sullivan, John Goerzen, Donald Bruce Stewart, O'Reilly Media, 2008 9. Prolog. Programowanie, W. F. Clocksin, C. S. Mellish, Helion, 2003 10. Praktyczny kurs Turbo Pascala. Wydanie IV, Tomasz M. Sadowski, Helion, 2003 11. Siedem języków w siedem tygodni. Praktyczny przewodnik nauki języków programowania, Tate, Helion, 2011
Witryna WWW modułu/przedmiotu	