

## KARTA PRZEDMIOTU

<b>Kod przedmiotu</b>	<b>0613-2INF-C30-PP</b>	
<b>Nazwa przedmiotu w języku</b>	polskim	<i>Paradygmaty programowania</i> <i>Programming Paradigms</i>
	angielskim	

### 1. USYTUOWANIE PRZEDMIOTU W SYSTEMIE STUDIÓW

<b>1.1. Kierunek studiów</b>	Informatyka
<b>1.2. Forma studiów</b>	Stacjonarne
<b>1.3. Poziom studiów</b>	Studia I-stopnia
<b>1.4. Profil studiów*</b>	Ogólnoakademicki
<b>1.5. Osoba przygotowująca kartę przedmiotu</b>	Tomasz Ruść
<b>1.6. Kontakt</b>	tomasz.rusc@ujk.edu.pl

### 2. OGÓLNA CHARAKTERYSTYKA PRZEDMIOTU

<b>2.1. Język wykładowy</b>	Polski
<b>2.2. Wymagania wstępne*</b>	Wstęp do programowania Podstawowa znajomość programowania w języku C,

### 3. SZCZEGÓŁOWA CHARAKTERYSTYKA PRZEDMIOTU

<b>3.1. Forma zajęć</b>	wykład, laboratorium, projekt	
<b>3.2. Miejsce realizacji zajęć</b>	Zajęcia w pomieszczeniu dydaktycznym UJK	
<b>3.3. Forma zaliczenia zajęć</b>	Wykłady – egzamin laboratorium – zaliczenie z oceną projekt - zaliczenie	
<b>3.4. Metody dydaktyczne</b>	Wykład, laboratoria- krótkie programy do napisania	
<b>3.5. Wykaz literatury</b>	<b>podstawowa</b>	<ol style="list-style-type: none"> <li>Joeren Fokker: Functional Programming. Department of Computer Science, Utrecht University 1995 (plik pdf dostępny w Internecie)</li> <li>Hal Daume III, et. al.: Yet Another Haskell Tutorial. 2004 (plik pdf dostępny w Internecie)</li> <li>Dave Stuart Robertson: Quick Prolog, <a href="http://www.dai.ed.ac.uk/groups/ssp/bookpages/quickprolog/quickprolog.html">http://www.dai.ed.ac.uk/groups/ssp/bookpages/quickprolog/quickprolog.html</a></li> <li>Patrick Blackburn, Johan Bos and Kristina Striegnitz: Learn Prolog Now!, <a href="http://www.learnprolognow.org/">http://www.learnprolognow.org/</a></li> </ol>
	<b>uzupełniająca</b>	<ol style="list-style-type: none"> <li>Kees Doets, Jan van Eijck: The Haskell Road to Logic, Math and Programming. 2004 (plik pdf dostępny w Internecie)</li> <li>W. Clocksin, C Mellish Prolog Programowanie Helion 2003</li> </ol>

### 4. CELE, TREŚCI I EFEKTY UCZENIA SIĘ

<p><b>4.1. Cele przedmiotu (z uwzględnieniem formy zajęć)</b></p> <p><b>Wykład:</b> C1 Celem przedmiotu jest zapoznanie studentów z czterema podstawowymi paradygmatami programowania: imperatywnym, obiektowym, funkcyjnym i deklaratywnym.</p> <p><b>Laboratorium:</b> C1 Nabycie umiejętności programowania w oparciu o cztery podstawowe paradygmatami.</p> <p><b>Projekt</b> C1 Dokumentowanie wykonanych czynności laboratoryjnych (wykonanie sprawozdań) C2 Doskonalenie umiejętności praktycznych w zakresie analizy kodu i obliczeń.</p>
<p><b>4.2. Treści programowe (z uwzględnieniem formy zajęć)</b></p>

**Wykłady**

Wprowadzenie do tematyki wykładu: wstępne porównanie podstawowych paradygmatów programowania. Programowanie funkcyjne w języku Haskell. Programowanie deklaratywne w języku PROLOG. Programowanie imperatywne – omówienie podstawowych zagadnień. Programowanie obiektowe – omówienie podstawowy zagadnień. Wykorzystanie różnych paradygmatów programowania do rozwiązywania problemów algorytmicznych. Przegląd innych paradygmatów programowania

**Laboratorium:**

1. Praca w Haskellu w trybie interaktywnym
2. Pisanie prostych programów w Haskellu wykorzystując: definiowane własne funkcji, przetwarzanie list, wykorzystując funkcje rekurencyjne
3. Pisanie prostych programów w Prologu wykorzystując: definiowanie faktów, reguł i zapytań, rekurencja, przetwarzanie list

**4.3. Przedmiotowe efekty uczenia się**

Efekt	Student, który zaliczył przedmiot	Odniesienie do kierunkowych efektów uczenia się
<b>w zakresie WIEDZY:</b>		
W01	posiada wiedzę w zakresie programowania funkcyjnego w języku Haskell	INF1A_W07
W02	posiada wiedzę w zakresie programowania w logice w języku Prolog	INF1A_W07
W03	posiada wiedzę dotyczącą różnych paradygmatów programowania i ich zastosowań	INF1A_W07
<b>w zakresie UMIEJĘTNOŚCI:</b>		
U01	potrafi czytać ze zrozumieniem, pisać, uruchamiać i weryfikować proste programy zapisane w języku programowania funkcyjnego (Haskell)	INF1A_U04 INF1A_U13
U02	potrafi czytać ze zrozumieniem, pisać, uruchamiać i weryfikować proste programy zapisane w języku programowania w logice (Prolog)	INF1A_U04 INF1A_U13
U03	potrafi określić, który z czterech podstawowych paradygmatów programowania (imperatywne, obiektowe, funkcyjne, deklaratywne) należy zastosować do danego problemu programistycznego	INF1A_U04 INF1A_U05 INF1A_U13
<b>w zakresie KOMPETENCJI SPOŁECZNYCH:</b>		
K01	ma świadomość ograniczeń wynikających z niedostatecznej wiedzy oraz umiejętności i rozumie potrzebę dalszego kształcenia	INF1A_K01

**4.4. Sposoby weryfikacji osiągnięcia przedmiotowych efektów uczenia się**

Efekty przedmiotowe (symbol)	Sposób weryfikacji (+/-)								
	Egzamin ustny/pisemny			Kolokwium			Praca własna		
	Forma zajęć			Forma zajęć			Forma zajęć		
	W	L	P	W	L	P	W	L	P
W01	+						+		
W02	+						+		
W03	+						+		
U01					+				
U02					+				
U03					+				
K01	+						+		

4.5. Kryteria oceny stopnia osiągnięcia efektów uczenia się		
Forma zajęć	Ocena	Kryterium oceny
wykład (W)	3	co najmniej 50% i nie więcej niż 60% łącznej liczby punktów możliwych do uzyskania
	3,5	ponad 60% i nie więcej niż 70% łącznej liczby punktów możliwych do uzyskania
	4	ponad 70% i nie więcej niż 80% łącznej liczby punktów możliwych do uzyskania
	4,5	ponad 80% i nie więcej niż 90% łącznej liczby punktów możliwych do uzyskania
	5	ponad 90% liczby punktów możliwych do uzyskania
Laboratorium(L)*	3	co najmniej 50% i nie więcej niż 60% łącznej liczby punktów możliwych do uzyskania
	3,5	ponad 60% i nie więcej niż 70% łącznej liczby punktów możliwych do uzyskania
	4	ponad 70% i nie więcej niż 80% łącznej liczby punktów możliwych do uzyskania
	4,5	ponad 80% i nie więcej niż 90% łącznej liczby punktów możliwych do uzyskania
	5	ponad 90% liczby punktów możliwych do uzyskania
Projekt (P)*	3	co najmniej 50% i nie więcej niż 60% łącznej liczby punktów możliwych do uzyskania
	3,5	ponad 60% i nie więcej niż 70% łącznej liczby punktów możliwych do uzyskania
	4	ponad 70% i nie więcej niż 80% łącznej liczby punktów możliwych do uzyskania
	4,5	ponad 80% i nie więcej niż 90% łącznej liczby punktów możliwych do uzyskania
	5	ponad 90% liczby punktów możliwych do uzyskania

#### 5. BILANS PUNKTÓW ECTS - NAKŁAD PRACY STUDENTA

Kategoria	Obciążenie studenta	
	Studia stacjonarne	Studia niestacjonarne
LICZBA GODZIN REALIZOWANYCH PRZY BEZPOŚREDNIM UDZIALE NAUCZYCIELA /GODZINY KONTAKTOWE/	75	
Udział w wykładach	30	
Udział w laboratoriach	30	
Realizacja projektu	15	
SAMODZIELNA PRACA STUDENTA /GODZINY NIEKONTAKTOWE/	75	
Przygotowanie do laboratorium	40	
Przygotowanie do egzaminu/kolokwium	35	
<b>ŁĄCZNA LICZBA GODZIN</b>	<b>150</b>	
<b>PUNKTY ECTS za przedmiot</b>	<b>6</b>	

**Przyjmuję do realizacji** (data i czytelne podpisy osób prowadzących przedmiot w danym roku akademickim)

.....