# KARTA OPISU MODUŁU KSZTAŁCENIA

| | | |
|---|---|---|
| Nazwa modułu/przedmiotu<br>**Programowanie wieloparadygmatowe** | | Kod<br>**1010334591010337136** |
| Kierunek studiów<br>**Informatyka** | Profil kształcenia (ogólnoakademicki, praktyczny)<br>**(brak)** | Rok / Semestr<br>**5 / 9** |
| Ścieżka obieralności/specjalność<br>**Technologie informatyczne** | Przedmiot oferowany w języku:<br>**polski** | Kurs (obligatoryjny/obieralny)<br>**obligatoryjny** |
| **Stopień studiów:**<br>**I stopień** | **Forma studiów** (stacjonarna/niestacjonarna)<br>**niestacjonarna** | |
| Godziny<br>Wykłady: **8** Ćwiczenia: **-** Laboratoria: **8** Projekty/seminaria: **-** | | Liczba punktów<br>**3** |
| Status przedmiotu w programie studiów (podstawowy, kierunkowy, inny)<br>**(brak)** | (ogólnouczelniany, z innego kierunku)<br>**(brak)** | |

| **Obszar(y) kształcenia i dziedzina(y) nauki i sztuki** | **Podział ECTS (liczba i %)** |
|---|---|
| **nauki techniczne** | **3 100%** |
| **nauki techniczne** | **3 100%** |

## Odpowiedzialny za przedmiot / wykładowca:

dr inż. Krzysztof Zwierzyński
email: Krzysztof.Zwierzynski@put.poznan.pl
tel. +48 61 665 3755
Wydział Elektryczny
ul. Piotrowo 3A 60-965 Poznań

## Wymagania wstępne w zakresie wiedzy, umiejętności, kompetencji społecznych:

| | | |
|---|---|---|
| 1 | **Wiedza:** | Student ma podstawowe wiadomości z zakresu logiki matematycznej, teorii funkcji rekurencyjnych, programowania imperatywnego i deklaratywnego, programowania zorientowanego obiektowo, baz danych, systemów operacyjnych oraz sieci komputerowych. |
| 2 | **Umiejętności:** | Student potrafi pozyskiwać informacje z literatury, baz danych i innych źródeł; potrafi integrować uzyskane informacje, dokonywać ich interpretacji, a także wyciągać wnioski oraz formułować opinie; posługuje się językiem angielskim w stopniu wystarczającym do porozumiewania się, a także do czytania ze zrozumieniem wskazanej literatury przedmiotowej. |
| 3 | **Kompetencje społeczne** | Student rozumie potrzebę i zna możliwości ciągłego dokształcania się, podnoszenia kompetencji językowych, zawodowych, osobistych i społecznych; ma świadomość odpowiedzialności za pracę własną oraz gotowość podporządkowania się zasadom pracy w zespole i ponoszenia odpowiedzialności za wspólnie realizowane zadania. |

## Cel przedmiotu:

przegląd paradygmatów obliczeniowych wykorzystywanych w informatyce ze szczególnym uwzględnieniem podstawowych pojęć, technik i abstrakcji programowych; wypracowanie umiejętności doboru modelu obliczeniowego do rozwiązywanego problemu; nabycie umiejętności programowania w wieloparadygmatowym środowisku obliczeniowym.

## Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia

### Wiedza:

1. Student ma uporządkowaną i podbudowaną teoretycznie wiedzę w zakresie podst. konstrukcji programistycznych, implementacji algorytmów, paradygmatów i stylów programowania, metod weryfikacji poprawności programów, języków formalnych, kompilatorów, platform - [K_W05]

### Umiejętności:

1. Student potrafi posłużyć się środowiskami i platformami programistycznymi do pisania, wykonywania i testowania prostych programów kodowanych w językach programowania imperatywnego, obiektowego i deklaratywnego - [K_U10]

### Kompetencje społeczne:

1. Student ma świadomość ważności dokładnego wykonania projektu, zachowania standardów notacyjnych, przestrzegania poprawności językowej i terminowego oddania prac - [K_K07]

## Sposoby sprawdzenia efektów kształcenia

Wykład. Pisemne sprawdzenie wiadomości obejmujące pytania teoretyczne oraz proste zadania.

Laboratorium. Ustna lub pisemna ocena przygotowania studentów do zajęć. Ocena aktywności studentów w zakresie realizacji bieżących ćwiczeń.

Warunki zaliczenia wykładu i laboratorium: należy uzyskać co najmniej 50% łącznej liczby punktów.

## Treści programowe

Wykład. Deklaratywny paradygmat obliczeniowy. Charakterystyczne pojęcia i techniki programowania funkcyjnego i deterministycznego programowanie w logice. Programowanie iteracyjne, rekurencyjne, metaprogramowanie, abstrakcyjne typy danych. Deklaratywna równoległość, czyli sterowany danymi model obliczeń równoległych. Relacyjny paradygmat obliczeniowy i bazy danych. Łączenie paradygmatów programowania w logice i programowania z więzami.

## Literatura podstawowa:

1. Armstrong J.: Programming Erlang. The Pragmatic Programmers, 2013

2. Haber F.:LEARN YOU SOME ERLANG FOR GREAT GOOD! A BEGINNER'S GUIDE (on-line learnyousomeerlang.com)

3. Niederliński A., Programowanie w logice z ograniczeniami. Łagodne wprowadzenie dla platformy ECLiPSe, Gliwice, 2014

4. Van Roy P., Haridi S.: Programowanie. Koncepcje, techniki i modele, Helion, Gliwice 2005

## Literatura uzupełniająca:

1. Brzykcy G., Meissner. A., Programowanie w Prologu i programowanie funkcyjne. Materiały do ćwiczeń, Wyd. PP., 1999

2. Cesarini F., Thompson S.: Erlang Programming. O'Reilly Media, 2009

3. Kowalski R., Logika w rozwiązywaniu zadań, WNT, Warszawa, 1989

4. Meissner A., Niwińska M., Zwierzyński K., Computing the Irregularity Strength of Connected Graphs by Parallel Constraint Solving in the Mozart System, Lecture Notes in Computer Science, Vol. 4967, Springer, Berlin-Heidelberg, 2008, s. 1096-1103.

5. Zwierzyński K.T., Meissner A., Niwińska M., A Method Involving Constraint Programming for Generating Integral Graphs without +-1 in the Spectrum. A Case Study, Studies in Automation and Information Technology, Vol. 35, PTPN, Poznań, 2010, s. 105-114.

## Bilans nakładu pracy przeciętnego studenta

| Czynność | Czas (godz.) |
|---|---|
| 1. Wykłady | 8 |
| 2. Laboratoria | 8 |
| 3. Przygotowanie do zajęć i egzaminu | 45 |

## Obciążenie pracą studenta

| forma aktywności | godzin | ECTS |
|---|---|---|
| Łączny nakład pracy | 75 | 3 |
| Zajęcia wymagające bezpośredniego kontaktu z nauczycielem | 30 | 1 |
| Zajęcia o charakterze praktycznym | 45 | 2 |