**karta przedmiotu**

1. Podstawowe informacje o przedmiocie

|  |  |
| --- | --- |
| Nazwa przedmiotu | Języki i paradygmaty programowania |
| Rocznik studiów | 2021/2022 |
| Kolegium | Informatyki Stosowanej |
| Kierunek studiów | Informatyka |
| Poziom kształcenia | Studia pierwszego stopnia - inżynierskie |
| Profil kształcenia | Praktyczny |
| Specjalność | - |
| Osoba odpowiedzialna | dr inż. Barbara Fryc |

1. Wymagania wstępne (wynikające z następstwa przedmiotów)

|  |
| --- |
| Matematyka, Algorytmy i struktury danych, Wstęp do programowania, Programowanie, Programowanie obiektowe |

1. Efekty uczenia się i sposób realizacji zajęć
   1. Cele przedmiotu

|  |  |
| --- | --- |
| C1 | Wykształcenie szerszego spojrzenia na programowanie, pokazanie cech wspólnych i różnic podstawowych języków programowania. |
| C2 | Pogłębienie znajomości języków programowania i zrozumienie podstawowych zagadnień implementacyjnych. |
| C3 | Kształcenie umiejętności programowania w języku funkcyjnym i języku programowania w logice. |
| C4 | Kształcenie umiejętności analizowania pod kątem teoretycznym kodu programów oraz umiejętności dowodzenia oraz podważania błędnych hipotez. |
| C5 | Kształcenie umiejętności prezentowania swoich działań na forum grupy oraz wyszukiwania i korzystania ze specjalistycznej literatury przedmiotu. |

* 1. Przedmiotowe efekty uczenia się, z podziałem na wiedzę, umiejętności i kompetencje społeczne, wraz z odniesieniem do efektów uczenia się dla kierunku

|  |  |  |
| --- | --- | --- |
| Lp. | Opis przedmiotowych efektów uczenia się | Odniesienie do efektów uczenia się dla kierunku |
| Po zaliczeniu przedmiotu student w zakresie **wiedzy** potrafi | | |
| P\_W01 | Wyjaśnić podstawowe pojęcia związane z językami programowania | K\_W01 |
| P\_W02 | Omówić cechy charakterystyczne języków programowania w czterech podstawowych paradygmatach oraz standardowe problemy programistyczne, które są za ich pomocą rozwiązywane |
| P\_W03 | Przeanalizować kod źródłowy programu i przewidzieć jego działanie. | K\_W03 |
| Po zaliczeniu przedmiotu student w zakresie **umiejętności** potrafi | | |
| P\_U01 | Tworzyć proste programy w języku PROLOG | K\_U16 |
| P\_U02 | Tworzyć proste programy w języku LISP |
| P\_U03 | Przygotowywać i przedstawić prezentację na podany temat z zakresu programowania w logice lub funkcyjnego i prowadzić dyskusję na forum grupy. | K\_U04 |
| Po zaliczeniu przedmiotu student w zakresie **kompetencji społecznych** | | |
| P\_K01 | Świadomie przekazuje informacje i opinie dotyczące osiągnięć informatyki, w szczególności zastosowania różnych języków programowania i robi to w sposób powszechnie zrozumiały. | K\_K06 |

* 1. Formy zajęć dydaktycznych oraz wymiar godzin i punktów ECTS

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Studia stacjonarne (ST) | | | | | | | |
| W | K | Ćw | L | ZP | P | eL | ECTS |
| 20 | - | - | 20 | - | 30 | - | 6 |
| Studia niestacjonarne (NST) | | | | | | | |
| W | K | Ćw | L | ZP | P | eL | ECTS |
| 16 | - | - | 14 | - | 20 | - | 6 |

* 1. Metody realizacji zajęć dydaktycznych

|  |  |
| --- | --- |
| Formy zajęć | Metoda realizacji |
| Wykład | Wykład informacyjno – problemowy. Prowadzący wykład wprowadza niezbędne pojęcia i omawia zagadnienia. W trakcie wykładu wykorzystuje prezentacje multimedialne oraz prezentuje programy napisane w poszczególnych językach programowania. Trudniejsze zagadnienia teoretyczne ilustrowane są przykładami i zadaniami, które są rozwiązywane ze studentami krok po kroku. Po każdym etapie rozwiązania stawiane są pytania i udzielanie odpowiedzi przez studentów oraz wykładowcę. Wykład omawia główne koncepcje związane z językami programowania i pokazuje przykłady z różnych języków. |
| Laboratorium | Zajęcia są realizowane metodami aktywnymi, z wykorzystaniem odpowiednio dobranych przykładów i zadań programistycznych (głównie w języku Lisp i Prolog). Na każdych zajęciach omawiane są wspólnie przykładowe zadania, a następnie studenci przy pomocy prowadzącego rozwiązują zadania z przygotowanej na każde zajęcia listy zadań. Studenci rozwiązują zadania zarówno pojedynczo jak i w przypadku trudniejszych zagadnień grupowo. Rozwiązania są omawiane na forum klasy przez pozostałych uczestników zajęć i prowadzącego zajęcia. Laboratoria skupiają się na poznaniu i zrozumieniu paradygmatów programowania funkcyjnego (Common Lisp) oraz programowania w logice (SWI-Prolog). |
| Projekt | Projektowa – indywidualna lub grupowa realizacja odpowiednio zdefiniowanego przedsięwzięcia programistycznego z zakresu programowania w logice lub programowania funkcyjnego. |

* 1. Treści kształcenia (oddzielnie dla każdej formy zajęć)

Projekt

|  |  |
| --- | --- |
| Lp. | Treści kształcenia realizowane w ramach projektu |
| P1 | Realizacja zadania projektowego polegającego na wykorzystaniu metod, technik i narzędzi programowania w logice lub programowania funkcyjnego. |

* 1. Korelacja pomiędzy efektami uczenia się, celami przedmiotu, a treściami kształcenia

|  |  |  |
| --- | --- | --- |
| Efekt kształcenia | Cele przedmiotu | Treści kształcenia |
| P\_W01 | C1, C2 | W1 – W8 |
| P\_W02 | C1, C2, C3 | W2-W8 |
| P\_W03 | C2-C4 | W3-W4, W6-W7 |
| P\_U01 | C3 | L1- L5 |
| P\_U02 | C3 | L6- L10 |
| P\_U03 | C3, C4, C5 | P1 |
| P\_K01 | C3, C4, C5 | P1 |

Wykład

|  |  |
| --- | --- |
| Lp. | Treści kształcenia realizowane w ramach wykładów |
|
| W1 | Historia języków programowania. Sposób opisu języków programowania. Opis składni i semantyki języków programowania. Rodzaje paradygmatów programowania. |
| W2 | Programowanie w logice: syntaktyka i semantyka rachunku predykatów, podejście deklaratywne vs. imperatywne, kluczowe elementy języka Prolog. |
| W3 | Programowanie w logice: przejście z reprezentacji problemu w rachunku predykatów na kod źródłowy w języku Prolog, zasada rezolucji. |
| W4 | Struktury danych w języku Prolog. |
| W5 | Programowanie funkcyjne: funkcje jako model programowania, rachunek lambda, dopasowywanie wzorca, nadawanie typów. |
| W6 | Programowanie funkcyjne: rekursja, leniwa ewaluacja, funkcje wyższego rzędu, przykłady z języków (Lisp). |
| W7 | Struktury danych w języku funkcyjnym (Lisp). |
| W8 | Paradygmat programowania obiektowego w różnych językach programowania. |

Ćwiczenia

|  |  |
| --- | --- |
| Lp. | Treści kształcenia realizowane w ramach ćwiczeń |
|
| L1 | PROLOG: wprowadzenie, obiekty i relacje, fakty, zapytania proste i złożone, zmienne. |
| L2 | PROLOG: reguły, omówienie składni języka, operatory, równość i unifikacja, obliczenia arytmetyczne, spełnianie celu oraz koniunkcji celów, nawracanie, unifikacja termów |
| L3 | PROLOG: struktury danych, listy (podstawowe algorytmy łączenie, dodawanie, znajdowanie elementu maksymalnego, długość listy, znajdowanie podlisty itp.), przeszukiwanie rekurencyjne, zastosowanie akumulatora |
| L4 | PROLOG: nawracanie i odcięcie, generowanie wielu rozwiązań, odcięcie (predykat fail), śledzenie i punkty kontrolne |
| L5 | PROLOG: przejście z rachunku predykatów na kod źródłowy w Prologu, postać klauzulowa, zapis klauzul, rezolucja i dowodzenie twierdzeń (klauzule Horna) |
| L6 | Omówienie środowiska Lispbox, LISP: elementy języka, EVAL, Set Quantity – setq i setf |
| L7 | LISP: notacja lambda, funkcje matematyczne i funkcje-podprogramy, mapowanie funkcji |
| L8 | LISP: funkcje logiczne, instrukcje warunkowe – IF, WHEN, UNLESS, CASE, COND |
| L9 | LISP: rekursja a iteracja, prezentacja standardowych zadań programistycznych dla języków funkcyjnych |
| L10 | LISP: listy – przydatne funkcje, porównanie kodu źródłowego w Lispie i Prologu |
|  | Zaliczenie przedmiotu – kolokwium |

* 1. Metody weryfikacji efektów uczenia się

|  |  |  |
| --- | --- | --- |
| Efekt  uczenia się | Metoda oceny | Forma zajęć, w ramach której następuje weryfikacja efektu |
| P\_W01 | Test otwarty | wykład |
| P\_W02 | Test otwarty |
| P\_W03 | Test otwarty |
| P\_U01 | Kolokwium | laboratorium |
| P\_U02 | Kolokwium |
| P\_U03 | Projekt | projekt |
| P\_K01 | Dyskusja | projekt |

* 1. Kryteria oceny stopnia osiągnięcia efektów uczenia się

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Efekt  uczenia się | Na ocenę 2  student nie potrafi | Na ocenę 3  student potrafi | Na ocenę 4  student potrafi | Na ocenę 5  student potrafi |
| P\_W01 | podać definicji danego pojęcia ani podać prostego przykładu ilustrującego dane pojęcie/zagadnienie | definiować proste pojęcia lub podać ich przykłady | definiować omawiane pojęcia z zakresu opisu języków programowania i podać ich przykłady | definiować omawiane pojęcia z zakresu opisu języków programowania i podać ich przykłady nieomawiane na wykładzie |
| P\_W02 | podać cech dwóch języków programowania zgodnych z różnymi paradygmatami | podać cechy dwóch języków programowania zgodnych z różnymi paradygmatami | podać cechy trzech języków programowania zgodnych z różnymi paradygmatami | podać cechy czterech języków programowania zgodnych z różnymi paradygmatami |
| P\_W03 | przewidzieć wynik działania programu | w sposób ogólny przewidzieć wynik działania programu | krok po kroku przeanalizować sposób działania programu oraz opisać działanie programu słownie | krok po kroku przeanalizować sposób działania programu oraz opisać działanie programu w sposób sformalizowany |
| P\_U01 | rozwiązać ani jednego zadania z trzech zaprezentowanych w szczególności związanego z utworzeniem prostej bazy wiedzy i skonstruowaniem do niej zapytań | rozwiązać jedno zadanie z trzech dostępnych polegające na utworzeniu prostej bazy wiedzy i skonstruowaniem do niej zapytań | rozwiązać dwa zadanie z trzech dostępnych: jedno związane z tworzeniem prostej bazy wiedzy, drugie z tworzeniem reguł i wykorzystywaniem operacji arytmetycznych | rozwiązać trzy zadanie w tym zadanie związane z rekurencyjnym przetwarzaniem list |
| P\_U02 | rozwiązać ani jednego zadania z trzech zaprezentowanych w szczególności związanego z definiowaniem funkcji matematycznych | rozwiązać jedno zadanie z trzech dostępnych polegające na definiowaniu funkcji matematycznych prostych i złożonych | rozwiązać dwa zadanie z trzech dostępnych: jedno związane z definicją funkcji matematycznych, drugie z wykorzystaniem instrukcji warunkowych i zagadnieniami iteracji i rekurencji | rozwiązać trzy zadanie w tym zadanie związane z przetwarzaniem list |
| P\_U03 | wystąpić na forum grupy i omówić rozwiązanie zadania | wystąpić na forum grupy i zaprezentować przynajmniej jedno zadanie z zakresu programowania w logice lub z zakresu programowania funkcyjnego z tematyk poznanej na laboratorium | wystąpić na forum grupy i zaprezentować przynajmniej dwa zadania, jedno zakresu programowania w logice, drugie z zakresu programowania funkcyjnego z tematyk poznanej na laboratorium | wystąpić na forum grupy i zaprezentować przynajmniej dwa zadania, jedno zakresu programowania w logice, drugie z zakresu programowania funkcyjnego z tematyk wykraczającej poza tematykę omawianą na laboratorium |
| P\_K01 | przekazać społeczeństwu informacje i opinie dotyczących osiągnięć informatyki, w szczególności zastosowania różnych języków programowania | w sposób dostateczny przekazać społeczeństwu informacje i opinie dotyczących osiągnięć informatyki, w szczególności zastosowania różnych języków programowania | w sposób dobry przekazać społeczeństwu informacje i opinie dotyczących osiągnięć informatyki, w szczególności zastosowania różnych języków programowania i robi to w sposób powszechnie zrozumiały. | doskonale przekazać społeczeństwu informacje i opinie dotyczących osiągnięć informatyki, w szczególności zastosowania różnych języków programowania i robi to w sposób powszechnie zrozumiały. |

* 1. Literatura

|  |
| --- |
| Literatura podstawowa |
| Clocksin, W. F., Mellish, C. S.: Prolog: programowanie, Helion, Gliwice 2003 lub nowsze |
| [Seibel, Peter](https://biblioteka.wsiz.rzeszow.pl/integro/search/description?q=Seibel%2C+Peter&index=3), Practical Common Lisp , [Berkeley, CA : Apress](https://biblioteka.wsiz.rzeszow.pl/integro/422100196142/seibel-peter/practical-common-lisp) 2005 lub nowsze |
| Prezentacje z wykładów |

|  |
| --- |
| Literatura uzupełniająca |
| R. Sebesta, Concepts of Programming Languages, Wydawnictwo Addison Wesley, 2008 lub nowsze |
| Norvig, Peter, Paradigms of artificial intelligence programming : case studies in Common Lisp , San Francisco, California : Morgan Kaufmann Publishers 1991 lub nowsze |
| M. Swaine: Programowanie funkcyjne, Wydawnictwo Naukowe PWN, 2020 lub nowsze |

1. Nakład pracy studenta - bilans punktów ECTS

|  |  |  |
| --- | --- | --- |
| **Rodzaje aktywności** | **Obciążenie studenta** | |
| **studia ST** | **studia NST** |
| Udział w W/K (UB) | 20 | 16 |
| Konsultacje do W/K (UB) | 4 | 3 |
| Udział w egzaminie z W (UB) | 2 | 2 |
| Samodzielne studiowanie tematyki W/K, w tym przygotowanie do egzaminu/zaliczenia | 6 | 11 |
| Udział w C/L (UB) | 20 | 14 |
| Konsultacje do C/L (UB) | 4 | 3 |
| Samodzielne przygotowanie się do C/L, w tym przygotowanie do zaliczenia | 26 | 33 |
| Udział w i konsultacje do PN(UB) | 30 | 20 |
| Samodzielne przygotowanie się do zaliczenia PN | 30 | 40 |
| **Sumaryczne obciążenie pracą studenta** | **142** | **142** |
| **Punkty ECTS za przedmiot** | **6** | **6** |
| **Punkty ECTS za zajęcia prowadzone z bezpośrednim udziałem nauczycieli i studentów (UB)** | **3** | **2** |
| **Punkty ECTS za zajęcia kształtujące umiejętności praktyczne (PZ)** | **0** | **0** |