**karta przedmiotu**

1. Podstawowe informacje o przedmiocie

|  |  |
| --- | --- |
| Nazwa przedmiotu | Języki i paradygmaty programowania |
| Rocznik studiów | 2022/2023 |
| 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 |
| 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** | | |
| P\_W01 | Potrafi wyjaśnić podstawowe pojęcia związane z językami programowania | K\_W01 |
| P\_W02 | Potrafi omówić cechy charakterystyczne języków programowania w czterech podstawowych paradygmatach oraz standardowe problemy programistyczne, które są za ich pomocą rozwiązywane | K\_W01 |
| P\_W03 | Potrafi przeanalizować kod źródłowy programu i przewidzieć jego działanie | K\_W03 |
| Po zaliczeniu przedmiotu student w zakresie **umiejętności** | | |
| P\_U01 | Potrafi tworzyć programy w języku Python | K\_U16 |
| P\_U02 | Potrafi tworzyć programy w języku F# | K\_U16 |
| P\_U03 | Potrafi przygotowywać i przedstawić prezentację na podany temat z zakresu programowania 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 Python i F#). 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 grupy przez pozostałych uczestników zajęć i prowadzącego zajęcia. |
| Projekt | Projektowa – indywidualna lub grupowa realizacja odpowiednio zdefiniowanego przedsięwzięcia programistycznego z zakresu programowania proceduralnego, obiektowego lub programowania funkcyjnego. |

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

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 funkcyjne: funkcje jako model programowania, rachunek lambda, dopasowywanie wzorca, nadawanie typów w F# |
| W3 | Programowanie funkcyjne: rekursja, leniwa ewaluacja, funkcje wyższego rzędu, przykłady z języków |
| W4 | Typy w programowaniu funkcyjnym |
| W5 | Wprowadzenie do imperatywnych kolekcji języka F# |
| W6 | Paradygmat programowania obiektowego w różnych językach programowania (Python, F#) |
| W7 | Refleksja w różnych językach programowania (F#) |
| W8 | Programowanie deklaratywne. Przykłady z różnych języków programowania |

Laboratorium

|  |  |
| --- | --- |
| Lp. | Treści kształcenia realizowane w ramach laboratorium |
| L1 | Programowanie proceduralne w Pythonie: zmienne lokalne, globalne, instrukcje sterujące, typy danych |
| L2 | Programowanie funkcyjne w Pythonie: rekurencja, operator lambda, funkcje anonimowe, [reguły obowiązywania zasięgu](https://rk.edu.pl/pl/funkcje-i-programowanie-funkcyjne/#1) |
| L3 | Programowanie funkcyjne w Pythonie, praca ze strukturami danych, Mapowanie i filtrowanie z wykorzystaniem list składanych, funkcja map(), reduce(), filter(), eval(), exec() |
| L4 | Programowanie obiektowe: dziedziczenie i hermetyzacja we Pythonie |
| L5 | F# - Nadawanie wartości i komunikacja z użytkownikiem. Liczby i łańcuchy znaków. Instrukcja let. Stosowanie instrukcji warunkowych i operatorów |
| L6 | F# - Niezmienne struktury danych |
| L7 | F# - Definiowanie funkcji rekurencyjnych |
| L8 | F# - tworzenie funkcji iteracyjnych |
| L9 | F# - praca z danymi tekstowymi |
| L10 | F# - programowanie z wykorzystaniem obiektów |

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 |

* 1. Metody weryfikacji efektów uczenia się

|  |  |  |
| --- | --- | --- |
| Efekt  uczenia się | Metoda oceny | Forma zajęć, w ramach której następuje weryfikacja efektu |
| P\_W01 | Pytania otwarte | Wykład |
| P\_W02 | Pytania otwarte |
| P\_W03 | Pytania otwarte |
| 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 prostego programy proceduralnego w języku Python | rozwiązać jedno zadanie z trzech dostępnych dotyczących zaganień dotyczacych paradygmatów programowania w języku Python | rozwiązać dwa zadanie z trzech dostępnych zaganień dotyczacych paradygmatów programowania w języku Python: jedno związane z tworzeniem programu proceduralnego, drugie z tworzeniem funkcji rekurencyjnych | rozwiązać trzy zadania w tym zadanie związane z tworzeniem obiektów w języku Python |
| P\_U02 | rozwiązać ani jednego zadania z trzech zaprezentowanych w szczególności związanego z utworzeniem prostego programy proceduralnego w języku F# | rozwiązać jedno zadanie z trzech dostępnych polegające na definiowaniu funkcji iteracyjnej i struktur danych w języku F# | rozwiązać dwa zadanie z trzech dostępnych: jedno związane z definicją funkcji iteracyjnej i struktur danych, drugie z wykorzystaniem funkcji rekurencyjnej | rozwiązać trzy zadanie w tym zadanie związane z tworzeniem obiektów w języku F# |
| P\_U03 | wystąpić na forum grupy i omówić rozwiązanie zadania | wystąpić na forum grupy i zaprezentować przynajmniej jedno zadanie z zakresu programowania proceduralnego lub z zakresu programowania funkcyjnego z tematyk poznanej na laboratorium | wystąpić na forum grupy i zaprezentować przynajmniej dwa zadania, jedno zakresu programowania proceduralnego, drugie z zakresu programowania funkcyjnego z tematyk poznanej na laboratorium | wystąpić na forum grupy i zaprezentować przynajmniej dwa zadania, jedno zakresu funkcyjnego proceduralnego, drugie z zakresu programowania obiektowego 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 |
| M. J. Kubiak: F#. Zadania z funkcyjnego i imperatywnego programowania z przykładowymi rozwiązaniami (ebook), Wyd. Helion, Gliwice 2020, lub nowsze |
| [M. Anaya](https://biblioteka.wsiz.rzeszow.pl/integro/search/description?q=Anaya%2C+Mariano&index=3): [Czysty kod w Pythonie. Twórz wydajny i łatwy w utrzymaniu kod](https://biblioteka.wsiz.rzeszow.pl/integro/423300666962/anaya-mariano/czysty-kod-w-pythonie?bibFilter=42), [Wyd. Helion](https://biblioteka.wsiz.rzeszow.pl/integro/search/description?q=Gliwice+%3A+Helion&index=1), Gliwice 2022, lub nowsze, dokument elektroniczny |
| Prezentacje z wykładów |

|  |
| --- |
| Literatura uzupełniająca |
| R. Sebesta: Concepts of Programming Languages, Wydawnictwo Addison Wesley, 2008 lub nowsze |
| P. Norvig: Paradigms of artificial intelligence programming. Case studies in Common Lisp, Morgan Kaufmann Publishers, San Francisco 1991, lub nowsze |
| M. Swaine: Programowanie funkcyjne, Wydawnictwo Naukowe PWN, Warszawa 2020, lub nowsze |
| D. Syme, A. Granicz, A. Cisternino: F# 4.0 dla zaawansowanych, Wyd. Helion, Gliwice 2017, 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 | 26 | 31 |
| 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** | **162** | **162** |
| **Punkty ECTS za przedmiot** | **6** | **6** |
| **Punkty ECTS za zajęcia prowadzone z bezpośrednim udziałem nauczycieli i studentów (UB)** | **3** | **2** |