**karta przedmiotu**

1. Podstawowe informacje o przedmiocie

|  |  |
| --- | --- |
| Nazwa przedmiotu | Języki programowania 1 |
| Rocznik studiów | 2022/2023 |
| Kolegium | Informatyki Stosowanej |
| Kierunek studiów | Informatyka |
| Poziom kształcenia | Studia pierwszego stopnia – licencjackie |
| Profil kształcenia | Praktyczny |
| Specjalność | Programowanie |
| Osoba odpowiedzialna | dr inż. Barbara Fryc |

1. Wymagania wstępne (wynikające z następstwa przedmiotów)

|  |
| --- |
| Algorytmy i struktury danych, Podstawy programowania |

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

|  |  |
| --- | --- |
| C1 | Pogłębienie znajomości języków programowania i zrozumienie podstawowych zagadnień implementacyjnych. |
| C2 | Kształcenie umiejętności implementacji programów z wykorzystaniem wiodących języków i narzędzi programistycznych. |
| C3 | Kształcenie umiejętności analizowania pod kątem teoretycznym kodu programów oraz umiejętności projektowania kodu. |
| C4 | 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 | Przeanalizować kod źródłowy programu i przewidzieć jego działanie | KW01 |
| P\_W02 | Wyjaśnić podstawowe pojęcia związane z językami programowania | KW03 |
| Po zaliczeniu przedmiotu student w zakresie **umiejętności** | | |
| P\_U01 | Wykonać implementację prostego zadania programistycznego. | K\_U16 |
| P\_U02 | Dokonać analizy niewielkiego problemu informatycznego, zaprojektować dla tego problemu program w języku wysokiego poziomu, wykonać implementację, usunąć błędy i przeprowadzić testy uruchomieniowe tego programu. | K\_U16 |
| Po zaliczeniu przedmiotu student w zakresie **kompetencji społecznych** | | |
| P\_U03 | Przygotowywać prezentację projektu i prowadzić dyskusję na forum grupy | K\_U04 |

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

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Studia stacjonarne (ST) | | | | | | | |
| W | K | Ćw | L | ZP | P | eL | ECTS |
|  |  |  | 32 |  | 24 |  | 7 |

* 1. Metody realizacji zajęć dydaktycznych

|  |  |
| --- | --- |
| Formy zajęć | Metoda realizacji |
| Laboratorium | Zajęcia realizowane są w laboratorium. Studenci otrzymują instrukcje na podstawie których wykonują ćwiczenia polegające na rozwiązywaniu problemów programistycznych przy wykorzystaniu wybranych metod oraz języków programowania. |
| Projekt | Projekt polega na zaprojektowaniu i implementacji aplikacji komputerowej w języku programowania wysokiego poziomu, przygotowaniu dokumentacji projektowej oraz prezentacji przedstawiającej wyniki pracy. Projekt realizowany jest w niewielkich zespołach lub samodzielnie, w zależności od złożoności problemu. |

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

Laboratorium

|  |  |
| --- | --- |
| Lp. | Treści kształcenia realizowane w ramach laboratorium |
| L1 | Zaawansowane programowanie w języku C#: Metody rozszerzające, typy generyczne, ograniczenia, delegaty, Func i Action, wyrażenia lambda, zdarzenia (events), wprowadzenie do refleksji. |
| L3 | Obsługa kolekcji w języku C# za pomocą LINQ. Interfejs IEnumerable, typy anonimowe, typy dynamiczne. |
| L4 | Podstawy programowania asynchronicznego, wielowątkowego i równoległego w języku C#. Biblioteka TPL. |
| L5 | Obsługa wyjątków, zarządzanie pamięcią, garbage collector, interfejs IDisposable, finalizery, destruktory. |
| L6 | Dostęp do bazy danych za pomocą Entity Framework. Podejście database-first oraz code-first. Wykorzystanie LINQ to Entites, interfejs IQueryable. |

Projekt

|  |  |
| --- | --- |
| Lp. | Treści kształcenia realizowane w ramach projektu |
| P1 | Przedmiotem projektu jest stworzenie niewielkiej aplikacji komputerowej w wybranym języku i technologii programowania. Temat realizowanej aplikacji jest ustalany indywidualnie z prowadzącym zajęcia. Zadanie zostanie podzielone na kilka etapów: specyfikacja problemu, projekt programu, implementacja i testowanie. Zrealizowany projekt musi być zaprezentowany przez zespół realizujący na forum grupy. |

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

|  |  |  |
| --- | --- | --- |
| Efekt uczenia się | Cele przedmiotu | Treści kształcenia |
| P\_W01 | C1, C3 | L1-L6 |
| P\_W02 | C1 | L1-L6 |
| P\_U01 | C2 | L1-L6 |
| P\_U02 | C2 | L1-L6, P1 |
| P\_U03 | C4 | P1 |

* 1. Metody weryfikacji efektów uczenia się

|  |  |  |
| --- | --- | --- |
| Efekt  uczenia się | Metoda oceny | Forma zajęć, w ramach której następuje weryfikacja efektu |
| P\_W01 | Kolokwium zaliczeniowe składające się z zadań polegających na analizie i interpretacji fragmentów kodu programu | Laboratorium |
| P\_W02 | Kolokwium zaliczeniowe z pytaniami otwartymi | Laboratorium |
| P\_U01 | Kolokwium zaliczeniowe składające się z prostych zadań programistycznych | Laboratorium |
| P\_U02 | Ocena dokumentacji projektowej oraz poprawności implementacji programu | Projekt |
| P\_U03 | Ocena prezentacji projektu | 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 | Przeanalizować przynajmniej 50% kodu źródłowego programu i przewidzieć jego działania | Przeanalizować przynajmniej 50% elementów kodu źródłowego programu i przewidzieć ich działanie | Przeanalizować przynajmniej 70% elementy kodu źródłowego programu i przewidzieć ich działanie | Przeanalizować przynajmniej 90% kodu źródłowego programu i przewidzieć ich działanie |
| P\_W02 | Wyjaśnić podstawowych pojęć związanych z językami programowania | Wyjaśnić w stopniu dostatecznym podstawowe pojęć związane z językami programowania | Wyjaśnić w stopniu dobrym podstawowe pojęć związane z językami programowania | Wyjaśnić w stopniu bardzo dobrym podstawowe pojęć związane z językami programowania |
| P\_U01 | Wykonać implementacji przynajmniej 50% zadań programistycznych zadanych w ramach kolokwium. | Wykonać implementację przynajmniej 50% zadań programistycznych zadanych w ramach kolokwium. | Wykonać implementację przynajmniej 70% zadań programistycznych zadanych w ramach kolokwium. | Wykonać implementację przynajmniej 90% zadań programistycznych zadanych w ramach kolokwium. |
| P\_U02 | Dokonać analizy niewielkiego problemu informatycznego oraz zaprojektować dla tego problemu programu w języku wysokiego poziomu. | Dokonać analizy niewielkiego problemu informatycznego o małym stopniu zaawansowania, zaprojektować dla tego problemu program w języku wysokiego poziomu, wykonać implementację, usunąć błędy i przeprowadzić testy uruchomieniowe tego programu. | Dokonać analizy niewielkiego problemu informatycznego o średnim stopniu zaawansowania, zaprojektować dla tego problemu program w języku wysokiego poziomu, wykonać implementację, usunąć błędy i przeprowadzić testy uruchomieniowe tego programu. | Dokonać analizy niewielkiego problemu informatycznego o dużym stopniu zaawansowania, zaprojektować dla tego problemu program w języku wysokiego poziomu, wykonać implementację, usunąć błędy i przeprowadzić testy uruchomieniowe tego programu. |
| P\_U03 | Przygotowywać prezentacji projektu i prowadzić dyskusji na forum grupy | Przygotowywać prezentację projektu | Przygotowywać prezentację projektu i przeprowadzić ją na forum grupy | Przygotowywać prezentację projektu, przeprowadzić ją, oraz prowadzić dyskusję na forum grupy |

* 1. Literatura

|  |
| --- |
| Literatura podstawowa |
| R. Miles, The C# Programming Yellow Book : Learn to Program in C# from First Principles, Independently Published, 2018 |
| J. Albahari, C# 8.0 in a Nutshell : The Definitive Reference, O'Reilly UK Limited, Farnham, 2020 |
| R. C. Martin, Clean Code : A Handbook of Agile Software Craftsmanship, Pearson Education, Upper Saddle River, 2009, lub nowsze wydanie |

|  |
| --- |
| Literatura uzupełniająca |
| I. Griffiths, Programming C# 8.0 : Build Windows, Web, and Desktop Applications, O'Reilly Media, Inc, Sebastopol, 2020 |

1. Nakład pracy studenta - bilans punktów ECTS

|  |  |
| --- | --- |
| **Rodzaje aktywności** | **Obciążenie studenta** |
| **studia ST** |
| Udział w C/L (UB) | 32 |
| Konsultacje do C/L (UB) | 10 |
| Samodzielne przygotowanie się do C/L, w tym przygotowanie do zaliczenia | 54 |
| Udział w i konsultacje do PS/PN/eL (UB) | 24 |
| Samodzielne przygotowanie się do zaliczenia PS/PN/eL | 48 |
| **Sumaryczne obciążenie pracą studenta** | **168** |
| **Punkty ECTS za przedmiot** | **7** |
| **Punkty ECTS za zajęcia prowadzone z bezpośrednim udziałem nauczycieli i studentów (UB)** | **3** |
| **Punkty ECTS za zajęcia kształtujące umiejętności praktyczne (PZ)** | **7** |