**karta przedmiotu**

1. Podstawowe informacje o przedmiocie

|  |  |
| --- | --- |
| Nazwa przedmiotu | Programowanie obiektowe |
| Rocznik studiów | 2021/2022 |
| Kolegium | Informatyki Stosowanej |
| Kierunek studiów | Informatyka |
| Poziom kształcenia | Studia pierwszego stopnia - inżynierskie |
| Profil kształcenia | Praktyczny |
| Specjalność | - |
| Osoba odpowiedzialna | dr Marek Jaszuk |

1. Wymagania wstępne (wynikające z następstwa przedmiotów)

|  |
| --- |
| Wstęp do programowania, Programowanie |

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

|  |  |
| --- | --- |
| C1 | Przekazanie wiedzy w zakresie metodyki i technik programowania obiektowego, w tym podstawowych technik algorytmicznych oraz znaczenia programowania obiektowego w różnych obszarach aktywności człowieka |
| C2 | Przekazanie wiedzy w zakresie technik, metod ewaluacji oraz testowania programów obiektowych |
| C3 | Rozwijanie znajomości i rozumienia metod, narzędzi, teorii i praktyk stosowanych do projektowania i implementacji oprogramowania w technice obiektowej z uwzględnieniem etapów określenia wymagań, specyfikacji, walidacji i testowania oprogramowania |
| C4 | Kształtowanie umiejętności pracy indywidualnej i w zespole porozumiewając się przy użyciu różnych kanałów komunikacji; umiejętności oszacowania czasu potrzebnego na realizację programu obiektowego; umiejętności opracowania harmonogramu i zrealizowania prac zapewniając dotrzymanie terminów |
| C5 | Kształtowanie umiejętności przeprowadzenia ewaluacji i weryfikacji programu obiektowego w kontekście ogólnych cech jakościowych i ilościowych uwzględniając istniejące ograniczenia |
| C6 | Kształtowanie umiejętności wykorzystywania narzędzi stosowanych w konstruowaniu i dokumentacji procesu wytwarzania programu obiektowego, ze szczególnym uwzględnieniem narzędzi do kontroli oprogramowania w tym rozproszonego systemu kontroli wersji |
| C7 | Kształtowanie umiejętności rozwiązywania postawionego zadania programistycznego, określania jego specyfikacji, zgodności z istniejącymi normami i standardami, oraz oceniania pozytywnych i negatywnych aspektów proponowanego rozwiązania |

* 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 | ma uporządkowaną wiedzę w zakresie metodyki i technik programowania obiektowego, w tym podstawowych technik algorytmicznych oraz znaczenia programowania obiektowego w różnych obszarach aktywności człowieka | K\_W03 |
| P\_W02 | ma wiedzę w zakresie technik, metod ewaluacji oraz testowania programów obiektowych | K\_W04 |
| P\_W03 | zna i rozumie metody, narzędzia, teorie i praktyki stosowane do projektowania i implementacji oprogramowania w technice obiektowej z uwzględnieniem etapów określenia wymagań, specyfikacji, walidacji i testowania oprogramowania | K\_W07 |
| Po zaliczeniu przedmiotu student w zakresie **umiejętności** | | |
| P\_U01 | potrafi pracować indywidualnie i w zespole porozumiewając się przy użyciu różnych kanałów komunikacji; umie oszacować czas potrzebny na realizację programu obiektowego; potrafi opracować harmonogram i zrealizować prace zapewniając dotrzymanie terminów | K\_U02 |
| P\_U02 | potrafi przeprowadzić ewaluację i weryfikację programu obiektowego w kontekście ogólnych cech jakościowych i ilościowych uwzględniając istniejące ograniczenia | K\_U07, K\_U17 |
| P\_U03 | potrafi efektywnie wykorzystywać narzędzia stosowane w konstruowaniu i dokumentacji procesu wytwarzania programu obiektowego, ze szczególnym uwzględnieniem narzędzi do kontroli oprogramowania w tym rozproszonego systemu kontroli wersji | K\_U09 |
| P\_U04 | potrafi zaproponować rozwiązanie postawionego zadania programistycznego porównując istniejące rozwiązania, określić jego specyfikację, zgodność z istniejącymi normami i standardami, oraz ocenić pozytywne i negatywne aspekty proponowanego rozwiązania | K\_U17 |

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

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Studia stacjonarne (ST) | | | | | | | |
| W | K | Ćw | L | ZP | P | eL | ECTS |
| 20 |  |  | 20 |  | 40 |  | 8 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Studia niestacjonarne (NST) | | | | | | | |
| W | K | Ćw | L | ZP | P | eL | ECTS |
| 16 |  |  | 16 |  | 30 |  | 8 |

* 1. Metody realizacji zajęć dydaktycznych

|  |  |
| --- | --- |
| Formy zajęć | Metoda realizacji |
| Wykład | Wykład informacyjny i problemowy, połączony z elementami demonstracji. |
| Laboratorium | Ćwiczenia laboratoryjne przy komputerze. W trakcie zajęć studenci dokonują samodzielnej implementacji i debugowania programów komputerowych z wykorzystaniem środowiska zintegrowanego. |
| Projekt | Zadaniem studentów jest zaprojektowanie i implementacja programu w technice obiektowej, zgodnego z ustaloną z prowadzącym specyfikacją. Projekt może być realizowany w niewielkich zespołach. W trakcie realizacji projektów studenci korzystają z narzędzi wspomagających kontrolę wersji, synchronizację realizacji zadań i komunikację w zespole, co ma odzwierciedlać warunki pracy zawodowej programistów. |

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

Wykład

|  |  |
| --- | --- |
| Lp. | Treści kształcenia realizowane w ramach wykładów |
| W1 | Wprowadzenie do tworzenia w środowisku zintegrowanym |
| W2 | Podstawy obiektowości – definiowanie klas, tworzenie obiektów, konstruktory, metody, pola danych, klonowanie obiektów |
| W3 | Dziedziczenie, metody wirtualne, polimorfizm |
| W4 | Abstrakcje – klasy i metody abstrakcyjne, interfejsy |
| W5 | Obsługa wyjątków |
| W6 | Zasady projektowania programów obiektowych – reguły SOLID |
| W7 | Tworzenie programów z interfejsem graficznym – technologie GUI |
| W8 | Przechowywanie danych – zapis danych do plików, komunikacja z bazą danych |
| W9 | Tworzenie specyfikacji wymagań programu |
| W10 | Testowanie programów obiektowych – testy jednostkowe |

Laboratorium

|  |  |
| --- | --- |
| Lp. | Treści kształcenia realizowane w ramach laboratorium |
| L1 | Wprowadzenie do zintegrowanego środowiska tworzenia programów. |
| L2 | Definiowanie klas i obiektów. |
| L3 | Dziedziczenie i polimorfizm |
| L4 | Definiowanie klas abstrakcyjnych i interfejsów. |
| L5 | Praca z kolekcjami. |
| L6 | Obsługa wyjątków. |
| L7 | Tworzenie programów z Graficznym Interfejsem użytkownika (GUI). |
| L8 | Operacje wejścia-wyjścia, praca z plikami i systemami bazodanowymi. |
| L9 | Wersjonowanie kodu i narzędzia komunikacyjne w pracy zespołowej. |

Projekt

|  |  |
| --- | --- |
| Lp. | Treści kształcenia realizowane w ramach projektu |
| P1 | Opracowanie specyfikacji programu. |
| P2 | Planowanie harmonogramu realizacji zadania programistycznego. |
| P3 | Projektowanie rozwiązania programistycznego |
| P4 | Praca z systemem kontroli wersji i narzędziami komunikacyjnymi. |
| P5 | Wykorzystanie bibliotek programistycznych. |
| P6 | Opracowanie dokumentacji projektu. |
| P7 | Przygotowanie prezentacji stworzonego rozwiązania. |

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

|  |  |  |
| --- | --- | --- |
| Efekt uczenia się | Cele przedmiotu | Treści kształcenia |
| P\_W01 | C1 | W1-W8 |
| P\_W02 | C2 | W6, W10 |
| P\_W03 | C3 | W1-W10 |
| P\_U01 | C4 | P1-P7 |
| P\_U02 | C5 | L1-L9, P7 |
| P\_U03 | C6 | P1-P7, L9 |
| P\_U04 | C7 | L1-L8, P1 |

* 1. Metody weryfikacji efektów uczenia się

|  |  |  |
| --- | --- | --- |
| Efekt  uczenia się | Metoda oceny | Forma zajęć, w ramach której następuje weryfikacja efektu |
| P\_W01 | Egzamin w formie testu otwartego | W |
| P\_W02 | Egzamin w formie testu otwartego | W |
| P\_W03 | Egzamin w formie testu otwartego | W |
| P\_U01 | Ocena projektu | P |
| P\_U02 | Kolokwium z laboratorium w formie zadań praktycznych | L |
| P\_U03 | Ocena projektu | P |
| P\_U04 | Kolokwium z laboratorium w formie zadań praktycznych | L |

* 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 | Zademonstrować wiedzy w zakresie metodyki i technik programowania obiektowego | Zademonstrować wiedzę w zakresie metodyki i technik programowania obiektowego | Zademonstrować wiedzę w zakresie metodyki i technik programowania obiektowego, w tym podstawowych technik algorytmicznych | Zademonstrować wiedzę w zakresie metodyki i technik programowania obiektowego, w tym podstawowych technik algorytmicznych oraz znaczenia programowania obiektowego w różnych obszarach aktywności człowieka |
| P\_W02 | Zademonstrować wiedzy w zakresie technik, metod ewaluacji oraz testowania programów obiektowych | Zademonstrować podstawową wiedzę w zakresie technik, metod ewaluacji oraz testowania programów obiektowych | Zademonstrować dobrą wiedzę w zakresie technik, metod ewaluacji oraz testowania programów obiektowych | Zademonstrować bardzo dobrą wiedzę w zakresie technik, metod ewaluacji oraz testowania programów obiektowych |
| P\_W03 | Zademonstrować znajomości i rozumienia metod, narzędzi, teorii i praktyk stosowanych do projektowania i implementacji oprogramowania w technice obiektowej | na poziomie podstawowym zademonstrować znajomość i rozumienie metod, narzędzi, teorii i praktyk stosowanych do projektowania i implementacji oprogramowania w technice obiektowej z uwzględnieniem etapów określenia wymagań, specyfikacji, walidacji i testowania oprogramowania | na poziomie dobrym zademonstrować znajomość i rozumienie metod, narzędzi, teorii i praktyk stosowanych do projektowania i implementacji oprogramowania w technice obiektowej z uwzględnieniem etapów określenia wymagań, specyfikacji, walidacji i testowania oprogramowania | na poziomie dobrym zademonstrować znajomość i rozumienie metod, narzędzi, teorii i praktyk stosowanych do projektowania i implementacji oprogramowania w technice obiektowej z uwzględnieniem etapów określenia wymagań, specyfikacji, walidacji i testowania oprogramowania |
|  |  |  |  |  |
| P\_U01 | pracować indywidualnie i w zespole porozumiewając się przy użyciu różnych kanałów komunikacji; oszacować czasu potrzebnego na realizację programu obiektowego; opracować harmonogramu i zrealizować prace zapewniając dotrzymanie terminów | na poziomie podstawowym: pracować indywidualnie i w zespole porozumiewając się przy użyciu różnych kanałów komunikacji; umie oszacować czas potrzebny na realizację programu obiektowego; potrafi opracować harmonogram i zrealizować prace zapewniając dotrzymanie terminów | na poziomie dobrym: pracować indywidualnie i w zespole porozumiewając się przy użyciu różnych kanałów komunikacji; umie oszacować czas potrzebny na realizację programu obiektowego; potrafi opracować harmonogram i zrealizować prace zapewniając dotrzymanie terminów | na poziomie bardzo dobrym: pracować indywidualnie i w zespole porozumiewając się przy użyciu różnych kanałów komunikacji; umie oszacować czas potrzebny na realizację programu obiektowego; potrafi opracować harmonogram i zrealizować prace zapewniając dotrzymanie terminów |
| P\_U02 | przeprowadzić ewaluacji i weryfikacji programu obiektowego w kontekście ogólnych cech jakościowych i ilościowych uwzględniając istniejące ograniczenia | na poziomie podstawowym przeprowadzić ewaluację i weryfikację programu obiektowego w kontekście ogólnych cech jakościowych i ilościowych uwzględniając istniejące ograniczenia | na poziomie dobrym przeprowadzić ewaluację i weryfikację programu obiektowego w kontekście ogólnych cech jakościowych i ilościowych uwzględniając istniejące ograniczenia | na poziomie bardzo dobrym przeprowadzić ewaluację i weryfikację programu obiektowego w kontekście ogólnych cech jakościowych i ilościowych uwzględniając istniejące ograniczenia |
| P\_U03 | wykorzystywać narzędzi stosowanych w konstruowaniu i dokumentacji procesu wytwarzania programu obiektowego, ze szczególnym uwzględnieniem narzędzi do kontroli oprogramowania w tym rozproszonego systemu kontroli wersji | na poziomie podstawowym wykorzystywać narzędzia stosowane w konstruowaniu i dokumentacji procesu wytwarzania programu obiektowego, ze szczególnym uwzględnieniem narzędzi do kontroli oprogramowania w tym rozproszonego systemu kontroli wersji | na poziomie dobrym wykorzystywać narzędzia stosowane w konstruowaniu i dokumentacji procesu wytwarzania programu obiektowego, ze szczególnym uwzględnieniem narzędzi do kontroli oprogramowania w tym rozproszonego systemu kontroli wersji | na poziomie bardzo dobrym wykorzystywać narzędzia stosowane w konstruowaniu i dokumentacji procesu wytwarzania programu obiektowego, ze szczególnym uwzględnieniem narzędzi do kontroli oprogramowania w tym rozproszonego systemu kontroli wersji |
| P\_U04 | zaproponować rozwiązania postawionego zadania programistycznego porównując istniejące rozwiązania, określić jego specyfikacji, zgodności z istniejącymi normami i standardami, oraz ocenić pozytywnych i negatywnych aspektów proponowanego rozwiązania | zaproponować rozwiązanie prostego zadania programistycznego porównując istniejące rozwiązania, określić jego specyfikację, zgodność z istniejącymi normami i standardami, oraz ocenić pozytywne i negatywne aspekty proponowanego rozwiązania | zaproponować rozwiązanie średniej złożoności zadania programistycznego porównując istniejące rozwiązania, określić jego specyfikację, zgodność z istniejącymi normami i standardami, oraz ocenić pozytywne i negatywne aspekty proponowanego rozwiązania | zaproponować rozwiązanie dużej złożoności zadania programistycznego porównując istniejące rozwiązania, określić jego specyfikację, zgodność z istniejącymi normami i standardami, oraz ocenić pozytywne i negatywne aspekty proponowanego rozwiązania |

* 1. Literatura

|  |
| --- |
| Literatura podstawowa |
| Herbert Schildt, Java : przewodnik dla początkujących, Gliwice : Wydawnictwo Helion, 2015 lub nowsze |
| Cay S. Horstmann, Gary Cornell, Java : podstawy, Helion, Gliwice, 2014 lub nowsze |
| Jacek Matulewski, C# : lekcje programowania : praktyczna nauka programowania dla platform .NET i .NET Core, Helion, Gliwice 2021 lub nowsze |
| Joseph Albahari, Eric Johannsen, C# 8.0 w pigułce, Helion, Gliwice, 2021 lub nowsze |

|  |
| --- |
| Literatura uzupełniająca |
| Herbert Schildt, Java : kompendium programisty, Gliwice : Wydawnictwo Helion, 2012 (lub nowsze). |
| Charlie Hunt, Binu John, Wydajność Javy, Gliwice : Helion, 2013 lub nowsze |
| Włodzimierz Gajda, Git : rozproszony system kontroli wersji, Gliwice : Wydawnictwo Helion, 2013 lub nowsze |
| J. Krawiec, JAVA. Programowanie obiektowe w praktyce, Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa, 2018 lub nowsze |
| B. Payne, Java, to takie proste, Wydawnictwo Naukowe PWN, Warszawa, 2018 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 | 16 |
| Konsultacje do C/L (UB) | 4 | 3 |
| Samodzielne przygotowanie się do C/L, w tym przygotowanie do zaliczenia | 26 | 31 |
| Udział w i konsultacje do PS/PN/eL (UB) | 40 | 30 |
| Samodzielne przygotowanie się do zaliczenia PS/PN/eL | 60 | 70 |
| **Sumaryczne obciążenie pracą studenta** | **182** | **182** |
| **Punkty ECTS za przedmiot** | **8** | **8** |
| **Punkty ECTS za zajęcia prowadzone z bezpośrednim udziałem nauczycieli i studentów (UB)** | **4** | **3** |
| **Punkty ECTS za zajęcia kształtujące umiejętności praktyczne (PZ)** | **6** | **6** |