

Karta opisu zajęć - Sylabus

Państwowa Wyższa Szkoła Techniczno-Ekonomiczna im. ks. Bronisława Markiewicza w Jarosławiu

I. INFORMACJE PODSTAWOWE

Nazwa zajęć: Programowanie obiektowe	Cykl kształcenia: 2021/2022	Data aktualizacji sylabusa: 01.10.2021
Nazwa kierunku studiów, poziom i profil kształcenia: Informatyka, studia pierwszego stopnia, profil praktyczny		
Język wykładowy: polski/angielski	Rodzaj zajęć: Przedmiot kierunkowy	
Rok studiów: 2	Semestr: 3	
Liczba punktów ECTS przypisana zajęciom: 6	Koordynator zajęć Imię, nazwisko, tytuł/stopień naukowy, adres e-mail: Janusz Bytnar, dr inż., janusz.bytnar@pwste.edu.pl	
Jednostka organizacyjna: Instytut Inżynierii Technicznej	Prowadzący zajęcia Imię, nazwisko, tytuł/stopień naukowy, adres e-mail: Janusz Bytnar, dr inż., janusz.bytnar@pwste.edu.pl	

FORMA PROWADZENIA ZAJĘĆ I LICZBA GODZIN

Ogólna liczba godzin zajęć dydaktycznych na studiach stacjonarnych i niestacjonarnych z podziałem na formy:

Studia stacjonarne		Studia niestacjonarne	
Wykład:	30	Wykład:	
Ćwiczenia:		Ćwiczenia:	
Laboratorium:	45	Laboratorium:	
Lektorat:		Lektorat:	
Projekt:		Projekt:	
Zajęcia praktyczne:		Zajęcia praktyczne:	
Seminarium:		Seminarium:	
Zajęcia terenowe:		Zajęcia terenowe:	
Praktyki:		Praktyki:	
Inna forma (jaka):		Inna forma (jaka):	
RAZEM:	75	RAZEM:	

II. INFORMACJE SZCZEGÓŁOWE

Wymagania wstępne i dodatkowe:

wymagania formalne - student powinien być zarejestrowany na semestr trzeci

wymagania wstępne - w zakresie Wiedzy: Student powinien posiadać wiedzę w zakresie realizowanym w ramach przedmiotu Algorytmy i struktury danych, Podstawy programowania.

w zakresie Umiejętności: student powinien posiadać podstawowe umiejętności w zakresie posługiwania się systemem komputerowym wyniesione z kształcenia na roku pierwszym w ramach przedmiotów: Algorytmy i struktury danych, Podstawy programowania,

wymagania w zakresie Kompetencji społecznych: Zrozumienie potrzeby ciągłego dokształcania się, umiejętność pozyskiwania informacji, umiejętność samokształcenia się.

Cel (cele) kształcenia dla zajęć:				
Celem kształcenia jest nabycie przez studentów wiedzy i umiejętności z zakresu programowania obiektowego, zapoznanie z metodologią i zasadami programowania obiektowego oraz współczesnym środowiskiem języka programowania obiektowego Microsoft Visual Studio.				
Efekty uczenia się określone dla zajęć				
Efekty uczenia się określone dla zajęć w kategorii wiedza, umiejętności oraz kompetencje społeczne oraz metody weryfikacji efektów uczenia się				
UWAGA:				
Dzielimy efekty uczenia się określone dla zajęć na kategorie wiedzy, umiejętności i kompetencji społecznych. Określone dla zajęć efekty uczenia się nie muszą obejmować wszystkich trzech kategorii i zależą od formy zajęć.				
Symbol efektów uczenia się określonego dla zajęć*	Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się, student w kategorii:			
Wiedzy - zna i rozumie				
E_01	zna składnię języka, struktury danych oraz potrafi objaśnić działanie instrukcji w wybranym języku programowania obiektowego			
E_02	zna i rozumie zastosowanie metodologii programowania obiektowego podczas rozwiązywania problemów informatycznych			
E_03	potrafi sformułować prosty algorytm oraz napisać program komputerowy rozwiązujący typowe zagadnienia obliczeniowe wykorzystując techniki programowania obiektowego			
Umiejętności - potrafi				
E_04	Wykonać projekt aplikacji zawierającej schemat klasy. Potrafi dodawać składniki klas w postaci pól i metod.			
E_05	Budować schematy klas wykorzystujących podstawowe zagadnienia programowania obiektowego jak abstrakcja, hermetyzacja, polimorfizm, dziedziczenie.			
E_06	Wykonać aplikacje okienkowe z graficznym interfejsem użytkownika korzystające z paradygmatów programowania obiektowego			
Kompetencji społecznych - jest gotów do				
E_07	potrafi działać w grupie, ma świadomość odpowiedzialności za pracę własną i całego zespołu, rozumie potrzebę systematycznej pracy w celu zdobywania wyższych kompetencji zawodowych.			
UWAGA!				
Zaleca się, aby w zależności od liczby godzin zajęć, liczba efektów uczenia się zawierała się w przedziale: 3-7, ale są to wartości umowne.				
TREŚCI PROGRAMOWE I ICH ODNIESIENIE DO EFEKTÓW UCZENIA SIĘ OKREŚLONYCH DLA ZAJĘĆ				
Treści programowe (uszczegółowione, zaprezentowane z podziałem na poszczególne formy zajęć, tj. wykład, ćwiczenia, laboratoria, projekty, seminaria i inne):				
Symbol treści programowych	Opis treści programowych	Forma zajęć	Liczba godzin	Odniesienie do efektów uczenia się określonych dla zajęć (symbol efektów uczenia się)
		wykład		

TP-01	<p>Omówienie paradygmatu programowania obiektowego – definiowanie programów za pomocą obiektów. Przedstawienie historii programowania obiektowego oraz cech programowania obiektowego.</p> <p>Pojęcie „Klasy” oraz „Obiektów” w porównaniu z typami podstawowymi. Podstawowe założenia paradygmatu obiektowego: Abstrakcja, Hermetyzacja, Polimorfizm, Dziedziczenie.</p>		3	E_01, E_02, E_03
TP-02	<p>Przedstawienie zasad programowania zorientowanego obiektowo w C++. Zasady definiowania Klas (składniki klasy, sekcje dostępu), typy definiowane przez użytkownika w postaci obiektów, metody jako funkcje składowe klas. Omówienie przykładów programów./ Characteristics of object-oriented programming (classes and their components, objects, basic types, hierarchies of classes, method - member functions). Examples of programs.</p>		4	E_01, E_02, E_03
TP-03	<p>Charakterystyka popularnych języków obiektowych. Zasady Hermetyzacji obiektów w C++ - składniki publiczne, prywatne, chronione. Zadania Konstruktorów oraz destruktorów w Klasach. Cechy konstruktora i destruktorów. Konstruktor domyślny. Omówienie przykładów programów.</p>		4	E_01, E_02, E_03

TP-04	<p>Przedstawienie możliwości ograniczeń dostępu do składowych klasy w C++ oraz w C#. Rola funkcji oraz klas zaprzyjaźnionych z innymi klasami. Przeladowanie operatorów (funkcja operatorowa, operatory jednoargumentowe, operatory dwuargumentowe). Zadania konstruktora kopiującego oraz operatora przypisania – omówienie problemów implementacji.</p>		4	E_01, E_02, E_03
TP-05	<p>Omówienie tematu wykorzystania klas na zasadzie Dziedziczenia. Pojęcie klasy bazowej i klasy pochodnej – omówienie możliwości dostępu do składników, dziedziczenie kilkupokoleniowe, wielokrotne, konwersje standardowe przy dziedziczeniu. Tryby dziedziczenia C++. Omówienie przykładów programów.</p>		4	E_01, E_02, E_03
TP-06	<p>Problemy dziedziczenia w C++. Hierarchie dziedziczenia. Zalety i wady dziedziczenia wielokrotnego. Wpływ dziedziczenia na hermetyzację danych. Omówienie mechanizmu dziedziczenia w innych językach obiektowych: C#, Java. Ograniczenia i problemy dziedziczenia klas. Omówienie możliwości oraz wykorzystania Funkcji wirtualnych. Omówienie przykładów programów.</p>		4	E_01, E_02, E_03
TP-07	<p>Przedstawienie zastosowania klas abstrakcyjnych w językach C++, C#. Destruktor wirtualny – przykłady zastosowań. Funkcje i klasy ogólne. Omówienie mechanizmu obsługi wyjątków w C++.</p> <p>Przedstawienie tematu operacji we/wy w C++ za pomocą strumieni. Operacje na plikach w C++.</p>		4	E_01, E_02, E_03

TP-08	<p>Uzupełnienie wiadomości związanych z różnicami języka C++ i C#.</p> <p>Podstawowe informacje na temat platformy programistycznej .NET (składniki platformy, rodzaje aplikacji, narzędzia dla programistów).</p> <p>Tworzenie graficznych interfejsów użytkownika – aplikacje w C#.</p> <p>Wykorzystanie formularzy – Windows Forms. Komponenty interfejsu graficznego użytkownika – właściwości i wykorzystanie. Zasady obsługi zdarzeń w aplikacjach Windows Okna dialogowe. Budowa menu.</p>		3	E_01, E_02, E_03
		laboratorium		
TP-09	<p>Tworzenie nowych projektów w środowisku programistycznym Microsoft Visual Studio. Podział kodu źródłowego na wiele plików. Wstępne przygotowanie kodu źródłowego poszczególnych klas. Tworzenie nowych obiektów - testowanie aplikacji.</p>		3	E_01, E_02, E_03
TP-10	<p>Rozbudowa aplikacji poprzez dodanie nowych metod. Definiowanie nowych metod. Pisanie definicji funkcji do których obiekty przekazywane są w formie argumentu. Edycja kodów źródłowych Konstruktorów i Destruktora w przykładowych klasach. Tworzenie Konstruktorów domyślnego. Uruchamianie i testowanie aplikacji</p>		7	E_01, E_02, E_03, E_04, E_05, E_07
TP-11	<p>Rozbudowa aplikacji poprzez dodanie w schemacie klasy mechanizmu funkcji zaprzyjaźnionej z klasą – testowanie aplikacji. Edycja kodów źródłowych konstruktora kopiującego i przedefiniowanie operatora przypisania – testowanie aplikacji.</p>		7	E_01, E_02, E_03, E_04, E_05, E_07

TP-12	Rozbudowa aplikacji poprzez dodanie plików kodu źródłowego z nowymi klasami dziedziczącymi. Definiowanie schematów nowych klas. Modyfikacja aplikacji poprzez zdefiniowanie w kodzie funkcji wirtualnej oraz destruktora wirtualnego – testowanie aplikacji.		7	E_01, E_02, E_03, E_04, E_05, E_07
TP-13	Tworzenie aplikacji Windows Forms – tworzenie prostych interfejsów użytkownika w C++ i C# – testowanie aplikacji.		7	E_01, E_02, E_03, E_04, E_05, E_06, E_07
TP-14	Tworzenie aplikacji okienkowej Windows Forms wykorzystującej schematy klas z poprzednich zajęć. Tworzenie interfejsów użytkownika oraz pisanie kodów źródłowych obsługi zdarzeń.		7	E_01, E_02, E_03, E_04, E_05, E_06, E_07
TP-15	Realizacja aplikacji Windows Forms w języku C# operującej na danych w postaci obiektów. Aplikacja wykorzystuje możliwości zapisu i odczytu informacji o obiektach w plikach tekstowych – prosta baza danych.		7	E_01, E_02, E_03, E_04, E_05, E_06, E_07

ZALECANA LITERATURA (w tym pozycje w języku obcym)

Literatura podstawowa (powinna być dostępna dla studenta w uczelnianej bibliotece):

1. Grębosz J.: „Symfonia C++ standard”, Edycja 2015
2. Eckel B.: „Thinking in C++”, Helion, Gliwice, 2006,
3. Matulewski J.: „Visual Studio 2017. Tworzenie aplikacji Windows w języku C#”, Helion, Gliwice, 2018,
4. Griffiths I.: „C# 8.0. Programowanie. Tworzenie aplikacji Windows, internetowych oraz biurowych”, Helion, Gliwice, 2020,

Literatura uzupełniająca:

1. Snell M., Power L. ; [tł. Tomasz Walczak]: „Microsoft Visual Studio 2010 : księga eksperta”, Helion 2011,
2. Boduch A.: „Wstęp do programowania w C#, Helion”, Gliwice, 2006
3. Troelsen S.: „Język C# i Platforma .NET”, Wydawnictwo Naukowe PWN, Warszawa 2006
4. Notatki z wykładu

III. INFORMACJE DODATKOWE

Odniesienie efektów uczenia się określonych dla zajęć i treści programowych do form zajęć i metod oceniania

Symbol efektu uczenia się określonego dla zajęć	Symbol treści programowych realizowanych w trakcie zajęć	Formy zajęć i metody dydaktyczne prowadzenia zajęć umożliwiające osiągnięcie założonych efektów uczenia się *	Metody weryfikacji osiągnięcia efektów uczenia się przypisanych do zajęć #
---	--	---	--

	Wiedza	wykład	
E_01	TP_01...TP_12	Wykład podający, ćwiczenia oparte na wykorzystaniu różnych źródeł wiedzy, studium przypadku	Zaliczenie ustne, test, projekt, Egzamin
E_02	TP_01... TP_12	Wykład podający, ćwiczenia oparte na wykorzystaniu różnych źródeł wiedzy, studium przypadku	Zaliczenie ustne, test, projekt, Egzamin
E_03	TP_01... TP_12	Wykład podający, ćwiczenia oparte na wykorzystaniu różnych źródeł wiedzy, studium przypadku	Zaliczenie ustne, test, projekt, Egzamin
	Umiejętności	ćwiczenia, laboratorium, projekt, zajęcia praktyczne	
E_04	TP_08...TP_12	Wykład podający, ćwiczenia oparte na wykorzystaniu różnych źródeł wiedzy, studium przypadku	Zaliczenie ustne, test, projekt, Egzamin
E_05	TP_08...TP_12	Wykład podający, ćwiczenia oparte na wykorzystaniu różnych źródeł wiedzy, studium przypadku	Zaliczenie ustne, test, projekt, Egzamin
E_06	TP_09...TP_12	Wykład podający, ćwiczenia oparte na wykorzystaniu różnych źródeł wiedzy, studium przypadku	Zaliczenie ustne, test, projekt, Egzamin
	Kompetencje społeczne	ćwiczenia, laboratorium, projekt, zajęcia praktyczne	
E_07	TP_08...TP_12	Wykład podający, ćwiczenia oparte na wykorzystaniu różnych źródeł wiedzy, studium przypadku	Zaliczenie ustne, test, projekt, Egzamin
<p>Metody weryfikacji osiągnięcia efektów uczenia się określonych dla zajęć, powinny być zróżnicowane w zależności od kategorii, tj. inne dla kategorii wiedza i inne dla kategorii umiejętności i kompetencje społeczne.</p> <p>Dla wykładu: * np. wykład podający, wykład problemowy, ćwiczenia oparte na wykorzystaniu różnych źródeł wiedzy # np. egzamin ustny, test, prezentacja, projekt Zaleca się podanie przykładowych zadań (pytań) służących weryfikacji osiągnięcia efektów uczenia się określonych dla zajęć.</p>			
BILANS PUNKTÓW ECTS			
OBCIĄŻENIE PRACĄ STUDENTA (godziny)			
Forma aktywności		Liczba godzin *	
Godziny zajęć (według harmonogramu) z nauczycielem akademickim lub inną osobą prowadzącą zajęcia		75	

Praca własna studenta		100 (przygotowanie do zajęć, czytanie literatury, przygotowanie do egzaminu)	
SUMA GODZIN:		175	
OBCIĄŻENIE PRACĄ STUDENTA (punkty ECTS)			
		Liczba punktów ECTS	
SUMARYCZNA LICZBA PUNKTÓW ECTS PRZYPIŚNANYCH DO ZAJĘĆ	Praca studenta wymagająca bezpośredniego kontaktu z nauczycielem akademickim lub inną osobą prowadzącą zajęcia	Ogółem: 6	2,6
	Praca własna studenta		3,4
* godziny lekcyjne, czyli 1 godz. oznacza 45 min;			
OPIS PRACY WŁASNEJ STUDENTA:			
Praca własna studenta musi być precyzyjnie opisana, uwzględniając charakter praktyczny zajęć. Należy podać symbol efektu uczenia się, którego praca własna dotyczy oraz metody weryfikacji efektów uczenia się stosowane w ramach pracy własnej. Przykładowe formy aktywności: (1) przygotowanie do zajęć, (2) opracowanie wyników, (3) czytanie wskazanej literatury, (4) napisanie raportu z zajęć, (5) przygotowanie do egzaminu, opracowanie projektu.			
Samodzielna praca związana z czytaniem wskazanej literatury oraz przygotowanie do egzaminu (E_01 – E_03). Samodzielna realizacja ćwiczeń programistycznych na podstawie instrukcji, przygotowanie mini-projektów indywidualnych oraz grupowych (E_04 – E_07).			
KRYTERIA OCENIANIA			
Zajęcia laboratoryjne kończą się zaliczeniem na ocenę, zaś przedmiot kończy się egzaminem. Na ocenę dostateczną student zrealizował projekty programistyczne dotyczące opisywanych treści programowych przedmiotu w zakresie podstawowym i potrafi omówić zastosowane techniki. Ponadto w dostatecznym stopniu potrafi wykorzystać przekazaną wiedzę podczas wykładu w realizowanych projektach programistycznych oraz podczas egzaminu pisemnego.			
Na ocenę dobrą student zrealizował kompletne projekty programistyczne dotyczące opisywanych treści programowych przedmiotu i potrafi omówić zastosowane techniki oraz uzasadnić ich wybór. Ponadto w dobrym stopniu potrafi wykorzystać przekazaną wiedzę podczas wykładu w realizowanych projektach programistycznych oraz podczas egzaminu pisemnego.			
Na ocenę bardzo dobrą student zrealizował kompletne projekty programistyczne dotyczące opisywanych treści programowych przedmiotu z zastosowaniem ponadstandardowych rozwiązań i potrafi omówić zastosowane techniki oraz uzasadnić ich wybór. Ponadto w bardzo dobrym stopniu potrafi wykorzystać przekazaną wiedzę podczas wykładu w realizowanych projektach programistycznych oraz podczas egzaminu pisemnego.			
INFORMACJA O PRZEWDYWANEJ MOŻLIWOŚCI WYKORZYSTANIA B-LEARNINGU			
istnieje			
INFORMACJA O PRZEWDYWANEJ MOŻLIWOŚCI WYKORZYSTANIA E-LEARNINGU			
istnieje			

.....
(data, podpis Koordynatora)

.....

odpowiedzialnego za zajęcia)

(data, podpis Dyrektora Instytutu/
Kierownika Jednostki Międzyinstytutowej)

.....
(data, podpis Kierownika Zakładu)

*Uwaga:
Karta opisu zajęć (sylabus) musi być dostępna dla studenta.*