

Karta opisu zajęć - Sylabus			
Państwowa Wyższa Szkoła Techniczno-Ekonomiczna im. ks. Bronisława Markiewicza w Jarosławiu			
I. INFORMACJE PODSTAWOWE			
Nazwa zajęć: Współczesne języki programowania		Cykl kształcenia rozpoczynający się w roku akademickim 2023/2024	
Nazwa kierunku studiów, poziom i profil kształcenia: Informatyka, studia pierwszego stopnia, profil praktyczny			
Język wykładowy: polski		Rodzaj zajęć: Przedmiot kierunkowy	
Rok studiów: 2		Semestr: 4	
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			
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:	15	Wykład:	
Ćwiczenia:		Ćwiczenia:	
Laboratorium:	45	Laboratorium:	
Lektorat:		Lektorat:	
Projekt:		Projekt:	
Zajęcia praktyczne:		Zajęcia praktyczne:	
Seminarium:		Seminarium:	
Zajęcia terenowe:		Zajęcia terenowe:	
Praktyki zawodowe:		Praktyki zawodowe:	
Inna forma (jaka):		Inna forma (jaka):	
RAZEM:	60	RAZEM:	
II. INFORMACJE SZCZEGÓŁOWE			
Wymagania wstępne i dodatkowe: w zakresie Wiedzy i Umiejętności: Student powinien posiadać wiedzę i umiejętności w zakresie realizowanym w ramach przedmiotu: „Podstawy programowania” oraz „Programowanie obiektowe”. wymagania w zakresie Kompetencji społecznych: Zrozumienie potrzeby ciągłego dokształcania się, umiejętność pozyskiwania informacji , umiejętność samokształcenia się.			
Cel (cele) kształcenia dla zajęć: Celem przedmiotu jest omówienie współczesnych technik programowania pojawiających się we współczesnych językach programowania: programowania strukturalne, obiektowe, funkcjonalne i innych na bazie wybranych języków programowania ze szczególnym uwzględnieniem języka Java i C#.			
EFEKTY UCZENIA SIĘ OKREŚLONE DLA ZAJĘĆ I ICH ODNIESIENIE DO EFEKTÓW UCZENIA SIĘ OKREŚLONYCH DLA KIERUNKU STUDIÓW			

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 one od formy zajęć.

Symbol efektów uczenia się określonego dla zajęć*	Treść efektu uczenia się. Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się, student w kategorii:	Odniesienie do efektów uczenia się określonych dla kierunku studiów (symbol efektów uczenia się)
Wiedzy - zna i rozumie		
E_01	Składnię i semantykę języków programowania, sposoby przekazywania parametrów do podprogramów, potrafi zdefiniować abstrakcyjne typy danych oraz wyjaśnić zasady przeciążania operatorów.	K_W06, K_W08
E_02	Zastosowanie metodologii programowania obiektowego podczas rozwiązywania problemów informatycznych.	K_W06, K_W08
E_03	Zastosowanie współczesnych języków programowania wysokiego poziomu, zna zasady doboru języka programowania do rozwiązywania problemów w zakresie oprogramowania sprzętu i usług; rozumie metody specyfikowania podstawowych wymagań w zakresie oprogramowania.	K_W06, K_W08, K_W12
Umiejętności - potrafi		
E_04	Pracować indywidualnie i w zespole; umie oszacować czas potrzebny na realizację zleconego zadania; potrafi opracować i zrealizować harmonogram prac zapewniający dotrzymanie terminów.	K_U01, K_U02, K_U08, K_U09, K_U12, K_U14, K_U15, K_U20
E_05	Pozyskiwać informacje z literatury, baz danych, i innych źródeł.	K_U01, K_U05, K_U06, K_U20
E_06	Swobodnie posługiwać się różnorodnymi środowiskami programistycznymi oraz różnymi technikami programowania.	K_U01, K_U05, K_U08, K_U09, K_U12, K_U14, K_U15, K_U20
Kompetencji społecznych - jest gotów do		
E_07	Służenia wsparciem i pomocą członkom zespołu w trudnych i stresujących sytuacjach. Rozumie i akceptuje potrzebę pracy w zespole.	K_K01, K_K03, K_K04

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 FORM ZAJĘĆ I METOD OCENIANIA

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	Metody dydaktyczne prowadzenia zajęć umożliwiające osiągnięcie założonych efektów uczenia się *	Metody weryfikacji osiągnięcia efektów uczenia się przypisanych do zajęć #
		wykład		
TP_01	Omówienie tematyki przedmiotu, literatury, form i zasad zaliczenia. Charakterystyka i porównanie współczesnych języków programowania (C++, C#, Java). Koncepcje języków programowania: programowanie strukturalne (proceduralne), programowanie obiektowe. Wykład z komentarzem i przykładami w języku C++, C# i Java.		Wykład podający, wykład problemowy - prezentacja, wykład z dyskusją	Zaliczenie końcowe - Test
TP_02	Wprowadzenie do programowania w języku Java. Wersje Javy. Aktualne edycje Javy. Narzędzia programistyczne. Wirtualna maszyna Javy oraz wieloplatformowość programów. Kompilacja kodów źródłowych i uruchamianie programów. Dynamiczne zarządzanie pamięcią. Wady Javy.		Wykład podający, wykład problemowy - prezentacja, wykład z dyskusją	Zaliczenie końcowe - Test

TP_03	Omówienie podstawowych elementów języka Javy (typy podstawowe, tablice, operatory, instrukcje warunkowe, pętle). Projektowanie programów w języku Java. Definiowanie Klas i tworzenie obiektów. Definiowanie metod oraz Konstruktorów. Związki między klasami - mechanizm dziedziczenia i kompozycji. Przykłady w języku Java.		Wykład podający, wykład problemowy - prezentacja, wykład z dyskusją	Zaliczenie końcowe - Test
TP_04	Metody wirtualne. Metody i klasy abstrakcyjne. Interfejsy oraz ich implementacja. Klasy wewnętrzne. Wykład z komentarzem i przykładami w języku Java.		Wykład podający, wykład problemowy - prezentacja, wykład z dyskusją	Zaliczenie końcowe - Test
TP_05	Programowanie graficznych interfejsów użytkownika w wersji AWT oraz Swing. Przegląd komponentów oraz ich właściwości. Okna i rozmieszczenie komponentów (managery rozkładu). Pojęcie Kontenera. Pojęcie Modalności. Różnice pomiędzy komponentami AWT i Swing. Przykłady w języku Java.		Wykład podający, wykład problemowy - prezentacja, wykład z dyskusją	Zaliczenie końcowe - Test

TP_06	Omówienie zaawansowanych komponentów Javy. Wykorzystanie w aplikacjach Javy Ramek, podpowiedzi, ikon, Menu rozwijalnych. Okna dialogowe, okna wyboru pliku i koloru. Listy rozwijalne, Drzewa. Wykład z komentarzem i przykładami w języku Java.		Wykład podający, wykład problemowy - prezentacja, wykład z dyskusją	Zaliczenie końcowe - Test
TP_07	Przedstawienie tematu obsługi zdarzeń w aplikacjach Javy. Tworzenie słuchacza. Interfejsy nasłuchu. Przyłączenie słuchacza do komponentów. Obsługa zdarzeń od myszki i przycisków. Przykłady w języku Java.		Wykład podający, wykład problemowy - prezentacja, wykład z dyskusją	Zaliczenie końcowe - Test
TP_08	Komponenty Swing i architektura Model-View-Controller. Modele i widoki. Komunikacja model – widok. Nasłuch zmian w modelach. Przykłady komponentów: Lista, Tabele, Drzewa. Wykład z komentarzem i przykładami w języku Java.		Wykład podający, wykład problemowy - prezentacja, wykład z dyskusją	Zaliczenie końcowe - Test

TP_09	Omówienie tematu WąTpów w Javie. Podstawowe pojęcia WąTpu i procesu. Tworzenie i uruchamianie wąTpów. Kończenie pracy wąTpu. Stany i synchronizacja wąTpów.		Wykład podający, wykład problemowy - prezentacja, wykład z dyskusją	Zaliczenie końcowe - Test
TP_10	Platforma tworzenia GUI – JavaFX. Różnice pomiędzy JavaFX a Swing. Definiowanie wyglądu aplikacji w języku XML. Zastosowanie arkuszy CSS. Narzędzie do tworzenia wyglądu GUI - Scene Builder. JavaFX – zarządcy rozkładu komponentów. Java FX - przykład projektu FXML.		Wykład podający, wykład problemowy - prezentacja, wykład z dyskusją	Zaliczenie końcowe - Test
		laboratorium		
TP_11	Zajęcia praktyczne – zapoznanie ze środowiskiem programistycznym NetBeans i Eclipse. Tworzenie nowych projektów. Pisanie kodów źródłowych w celu powtórzenia podstawowych typów danych, funkcji, pętli, tablic jedno i dwu wymiarowych. Przekazywanie parametrów do funkcji. Przykładowe programy w Javie – kompilacja i testowanie.		Zajęcia praktyczne oparte na wykorzystaniu różnych źródeł wiedzy - praktyczna realizacja aplikacji	kolokwium, projekt zaliczeniowy

TP_12	Programowanie obiektowe w Javie. Zastosowanie klas, jako abstrakcyjnych typów danych. Przykładowa realizacja programów w języku C# i Java. Rola dziedziczenia w programowaniu obiekowym (Java). Polimorfizm w klasach. Realizacja i implementacja interfejsów. Przykłady w języku C# i Java.		Zajęcia praktyczne oparte na wykorzystaniu różnych źródeł wiedzy - praktyczna realizacja aplikacji	kolokwium, projekt zaliczeniowy
TP_13	Zajęcia praktyczne z tworzenie interfejsów graficznych użytkownika z wykorzystaniem AWT i Swing. Ręczne i automatyczne rozmieszczanie komponentów na formatce. Menedżery rozkładu komponentów w panelach i oknach ramowych. Wykorzystanie klasy JFrame. Pisanie programów w języku Java.		Zajęcia praktyczne oparte na wykorzystaniu różnych źródeł wiedzy - praktyczna realizacja aplikacji	kolokwium, projekt zaliczeniowy
TP_14	Szybkie tworzenie GUI za pomocą Swing GUI Forms - wykorzystanie komponentów Swingowych w aplikacjach i apletach (wykorzystanie klasy JApplet). Okna dialogowe, obsługa zdarzeń podstawowych komponentów. Pisanie programów w języku Java.		Zajęcia praktyczne oparte na wykorzystaniu różnych źródeł wiedzy - praktyczna realizacja aplikacji	kolokwium, projekt zaliczeniowy

TP_15	Pisanie aplikacji w języku Java z zaawansowanymi komponentami (wykorzystanie klasy JOptionpane, FileDialog). Odczytywanie i zapisywanie plików – wykorzystanie klas FileOutputStream i FileInputStream. Tworzenie menu i list rozwijalnych.		Zajęcia praktyczne oparte na wykorzystaniu różnych źródeł wiedzy - praktyczna realizacja aplikacji	kolokwium, projekt zaliczeniowy
TP_16	Zajęcia praktyczne – instalacja Scene Buildera i konfiguracja w NetBeans. Tworzenie aplikacji JavaFX z wykorzystaniem SceneBuildera.		Zajęcia praktyczne oparte na wykorzystaniu różnych źródeł wiedzy - praktyczna realizacja aplikacji	kolokwium, projekt zaliczeniowy
TP_17	Pisanie zaawansowanej aplikacji w Javie korzystającej z bazy danych SQLite na licencji Public Domain.		Zajęcia praktyczne oparte na wykorzystaniu różnych źródeł wiedzy - praktyczna realizacja aplikacji	kolokwium, projekt zaliczeniowy

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.

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.

ZALECANA LITERATURA (w tym pozycje w języku obcym)

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

1. Schildt H.: „Java. Kompendium programisty. Wydanie X”, Helion, Gliwice, 2019,
2. Eckel B.: „Thinking in Java”, Helion, Gliwice, 2006,
3. Bochenek A.: „Eclipse : biblioteka SWT”, wyd. Mikom, 2008,
4. Bloch J.: „Java. Efektywne programowanie. Wydanie III, Helion, Gliwice, 2018

Literatura uzupełniająca:

1. Schildt H.: „Java. Przewodnik dla początkujących. Wydanie VIII”, Helion, Gliwice, 2020,
2. Heffelfinger D.: „Java EE 6. Tworzenie aplikacji w NetBeans 7”, Helion, Gliwice, 2014,
3. Notatki z wykładu

III. INFORMACJE DODATKOWE

BILANS PUNKTÓW ECTS

OBCIĄŻENIE PRACĄ STUDENTA (godziny)			
Forma aktywności		Liczba godzin *	
Godziny zajęć (według harmonogramu) z nauczycielem akademickim lub inną osobą prowadzącą zajęcia		60	
Praca własna studenta		60(czytanie literatury, praca nad realizacją projektu)	
SUMA GODZIN:		120	
OBCIĄŻENIE PRACĄ STUDENTA (punkty ECTS)			
		Liczba punktów ECTS	
SUMARYCZNA LICZBA PUNKTÓW ECTS PRZYPISANYCH DO ZAJĘĆ	Praca studenta wymagająca bezpośredniego kontaktu z nauczycielem akademickim lub inną osobą prowadzącą zajęcia	Ogółem: 6	3
	Praca własna studenta		3
* 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 symbole 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 zaliczenia końcowego (E_01 – E_03). Samodzielna realizacja ćwiczeń programistycznych na podstawie instrukcji, przygotowanie mini-projektów indywidualnych oraz grupowych (E_04 – E_07).			
KRYTERIA OCENIANIA			
Zajęcia kończą się zaliczeniem na ocenę. 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 zaliczenia 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 zaliczenia 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 zaliczenia pisemnego.			
INFORMACJA O PRZEWIDYWANEJ MOŻLIWOŚCI WYKORZYSTANIA KSZTAŁCENIA NA ODLEGŁOŚĆ			

.....
(data, podpis Koordynatora
odpowiedzialnego za zajęcia)

.....
(data, podpis Kierownika Zakładu/
Kierownika Jednostki Międzyinstytutowej)

Uwaga:

Karta opisu zajęć (syllabus) musi być dostępna dla studenta.