

## Karta opisu zajęć - Sylabus

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

### I. INFORMACJE OGÓLNE

Nazwa zajęć: Współczesne języki programowania		Kod zajęć: C18	
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: 5	Data aktualizacji sylabusu: 01.10.2021 r.
Instytut (Zakład) odpowiedzialny za zajęcia:		Instytut Inżynierii Technicznej	
Imię, nazwisko, tytuł/stopień naukowy, adres e-mail wykładowcy (wykładowców)/prowadzących zajęcia:			

### 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:	40	Praktyki:	
Inna forma (jaka):		Inna forma (jaka):	
<b>RAZEM:</b>	100	<b>RAZEM:</b>	

### II. INFORMACJE SZCZEGÓLNE

Cel (cele) prowadzenia 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#.

Wymagania wstępne w zakresie wiedzy, umiejętności oraz kompetencji społecznych (jeśli obowiązują):

Wymagania wstępne -

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.

Przypisane do zajęć efekty uczenia się w zakresie wiedzy, umiejętności oraz kompetencji społecznych i odniesienie ich do efektów uczenia się dla określonego kierunku studiów, poziomu i profilu.

**UWAGA:**

Dzielimy efekty uczenia się przypisane do zajęć na kategorie wiedzy, umiejętności i kompetencji społecznych. Przypisane do zajęć efekty uczenia się nie muszą obejmować wszystkich trzech kategorii.

Symbol efektów uczenia się przypisanego do zajęć*	Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się, student w kategorii:	Odniesienie do efektów uczenia się dla określonego kierunku studiów, poziomu i profilu #
<b>Wiedzy - zna i rozumie</b>		
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
<b>Umiejętności - potrafi</b>		
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
<b>Kompetencji społecznych - jest gotów do</b>		
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
<p>* kod zajęć,  # efekty uczenia się dla określonego kierunku studiów, poziomu i profilu (np. K_W01, K_U01, ..)  W- wiedza, U- umiejętności, K- kompetencje społeczne  01, 02...- numer efektu uczenia się</p> <p><b>UWAGA!</b>  Zaleca się, aby w zależności od liczby godzin zajęć, liczba efektów uczenia się zawierała się w przedziale: 3-7, ale są to wartości umowne w zależności od ogólnej liczby godzin zajęć.</p>		
<b>TREŚCI PROGRAMOWE I ICH ODNIESIENIE DO EFEKTÓW UCZENIA SIĘ PRZYPISANYCH DO ZAJĘĆ</b>		

Symbol treści programowych	Opis treści programowych	Liczba godzin	Odniesienie do efektów uczenia się przypisanych do zajęć
<b>wykład</b>			
TK_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.	2	E_01
TK_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.	1	E_01, E_02, E_05
TK_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.	2	E_01, E_02, E_05
TK_04	Metody wirtualne. Metody i klasy abstrakcyjne. Interfejsy oraz ich implementacja. Klasy wewnętrzne. Wykład z komentarzem i przykładami w języku Java.	1	E_01, E_02, E_05
TK_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.	2	E_01, E_02, E_05
TK_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.	1	E_01, E_02, E_03, E_05
TK_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.	2	E_01, E_02, E_03, E_05
TK_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.	1	E_01, E_02, E_03, E_05
TK_09	Omówienie tematu Wątków w Javie. Podstawowe pojęcia Wątku i procesu. Tworzenie i uruchamianie wątków. Kończenie pracy wątku. Stany i synchronizacja wątków.	1	E_01, E_02, E_03, E_05

TK_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.	2	E_01, E_02, E_03, E_05
<b>ćwiczenia</b>			
<b>laboratorium</b>			
TK_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.	6	E_01, E_02, E_03, E_04, E_05, E_06, E_07
TK_12	Programowanie obiektowe w Javie. Zastosowanie klas, jako abstrakcyjnych typów danych. Przykładowa realizacja programów w języku C# i Java. Rola dziedziczenia w programowaniu obiektowym (Java). Polimorfizm w klasach. Realizacja i implementacja interfejsów. Przykłady w języku C# i Java.	7	E_01, E_02, E_03, E_04, E_05, E_06, E_07
TK_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.	6	E_01, E_02, E_03, E_04, E_05, E_06, E_07
TK_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.	6	E_01, E_02, E_03, E_04, E_05, E_06, E_07
TK_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.	6	E_01, E_02, E_03, E_04, E_05, E_06, E_07
TK_16	Zajęcia praktyczne – instalacja Scene Buildera i konfiguracja w NetBeans. Tworzenie aplikacji JavaFX z wykorzystaniem SceneBuildera.	6	E_01, E_02, E_03, E_04, E_05, E_06, E_07
TK_17	Pisanie zaawansowanej aplikacji w Javie korzystającej z bazy danych SQLite na licencji Public Domain.	8	E_01, E_02, E_03, E_04, E_05, E_06, E_07
<b>seminarium</b>			
ZALECANA LITERATURA (w tym pozycje w języku obcym)			

**Literatura podstawowa przedmiotu** (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 przedmiotu:**

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

INFORMACJA O PRZEWIDYWANEJ MOŻLIWOŚCI WYKORZYSTANIA B-LEARNINGU

INFORMACJA O PRZEWIDYWANEJ MOŻLIWOŚCI WYKORZYSTANIA E-LEARNINGU

### III. INFORMACJE DODATKOWE

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

**Metody weryfikacji osiągnięcia efektów uczenia się przypisanych do zajęć, powinny być zróżnicowane w zależności od kategorii, tj. inne dla kategorii wiedza i inne dla kategorii umiejętności i kompetencje społeczne.**

\* 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 przypisanych do zajęć.

Symbol efektu uczenia się przypisanego do 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 #
<b>Wiedza</b>			
E_01	TK_01...TK_17	Wykład i zajęcia praktyczne – prezentacja, wykład z dyskusją, praktyczna realizacja aplikacji	Test, kolokwium, projekt zaliczeniowy
E_02	TK_02...TK_17	Wykład i zajęcia praktyczne – prezentacja, wykład z dyskusją, praktyczna realizacja aplikacji	Test, kolokwium, projekt zaliczeniowy
E_03	TK_06...TK_17	Wykład i zajęcia praktyczne – prezentacja, wykład z dyskusją, praktyczna realizacja aplikacji	Test, kolokwium, projekt zaliczeniowy
<b>Umiejętności</b>			
E_04	TK_11...TK_17	Wykład i zajęcia praktyczne – prezentacja, wykład z dyskusją, praktyczna realizacja aplikacji	Test, kolokwium, projekt zaliczeniowy
E_05	TK_02...TK_17	Wykład i zajęcia praktyczne – prezentacja, wykład z dyskusją, praktyczna realizacja aplikacji	Test, kolokwium, projekt zaliczeniowy

E_06	TK_11...TK_17	Wykład i zajęcia praktyczne – prezentacja, wykład z dyskusją, praktyczna realizacja aplikacji	Test, kolokwium, projekt zaliczeniowy
<b>Kompetencje społeczne</b>			
E_07	TK_11...TK_17	Wykład podający, ćwiczenia oparte na wykorzystaniu różnych źródeł wiedzy, studium przypadku	Test, kolokwium, projekt zaliczeniowy
<b>MIARA ŚREDNIEGO NAKŁADU PRACY STUDENTA NIEZBĘDNA DO UZYSKANIA EFEKTÓW UCZENIA SIĘ (godziny)</b>			
Forma aktywności		Średnia liczba godzin na zrealizowanie aktywności *	
Godziny zajęć (według harmonogramu) z nauczycielem		60	
w tym liczba godzin z praktyk zawodowych realizowanych w uczelni (według harmonogramu)		40	
Praca własna studenta #		60( czytanie literatury, praca nad realizacją projektu)	
<b>SUMA GODZIN:</b>		<b>160</b>	
<b>MIARA ŚREDNIEGO NAKŁADU PRACY STUDENTA NIEZBĘDNA DO UZYSKANIA EFEKTÓW UCZENIA SIĘ (punkty ECTS)</b>			
			Liczba punktów ECTS *
SUMARYCZNA LICZBA PUNKTÓW ECTS PRZYPISANYCH DO ZAJĘĆ	Praca studenta wymagająca bezpośredniego kontaktu z nauczycielem akademickim		3
	Liczba punktów ECTS przypisana praktykom zawodowym, jeśli formą zajęć dla tego przedmiotu są praktyki zawodowe		
	Praca własna studenta		2
* godziny lekcyjne, czyli 1 godz. oznacza 45 min.			
# przykładowe formy aktywności: (1) przygotowanie do zajęć, (2) opracowanie wyników, (3) czytanie wskazanej literatury, (4) napisanie raportu z zajęć, (5) przygotowanie do egzaminu,...			
<b>KRYTERIA OCENIANIA I WYMAGANIA EGZAMINACYJNE</b>			
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.			
<b>Kryteria różnicowania ocen w powiązaniu ze stopniem realizacji efektów uczenia się, muszą być: precyzyjne i czytelne.</b>			

**Podpis nauczyciela akademickiego lub  
osoby odpowiedzialnej za przedmiot:**

.....  
(imię i nazwisko)

.....  
( podpis i data)

**Podpis kierownika zakładu:**

.....  
(imię i nazwisko)

.....  
( podpis i data)

**Podpis dyrektora instytutu:**

.....  
(imię i nazwisko)

.....  
( podpis i data)