

<b>Nazwa zajęć:</b> Języki i paradygmaty programowania				Programming languages and paradigms	
<b>Kierunek:</b> Informatyka				<b>Obowiązuje od roku ak.</b> 2019/2020	
<b>Poziom:</b> I st. inżynierski		<b>Profil:</b> praktyczny		<b>Grupa zajęć:</b> Kierunkowe	
<b>Semestr:</b> III	<b>Forma zaliczenia:</b> E - egzamin	<b>Punkty ECTS:</b> 6		<b>Zajęcia do wyboru:</b> Nie	<b>Język zajęć:</b> polski

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych:

<b>Wykład</b> 15 / 8	<b>Ćwiczenia</b> 45 / 24	<b>Projekt</b> 15 / 8	<b>Suma godzin:</b> 75 / 40
-------------------------	-----------------------------	--------------------------	--------------------------------

**Specjalność:**

**Nazwiska osób odpowiedzialnych za zajęcia:**

mgr Marcin Zawadzki

**Opis zajęć:**

Przedmiot "Języki i paradygmaty programowania" przeznaczony jest dla studentów kierunku Informatyka, którzy przynajmniej w podstawowym zakresie posiadają wiedzę na temat programowania między innymi w takich językach jak: C++, Java, Python itp.. Programowanie jest dziedziną wymagającą abstrakcyjnego, logicznego oraz analitycznego myślenia. Celem przedmiotu jest wykształcenie umiejętności sprawnego posługiwania się dowolnym językiem programowania w sposób, który możliwie najlepiej odzwierciedli rzeczywistość. Całość kursu nastawiona jest na możliwie najprostsze przekazanie wiedzy o tym, czym metody, funkcje, instrukcje warunkowe, pętle itp. oraz jakie w praktyczny sposób wykorzystać zdobytą wiedzę.

**Cele dydaktyczne:**

Zapoznanie studentów z zagadnieniami dotyczącymi programowania.

Nabywanie przez studenta wiedzy o podstawowych zagadnieniach takich jak instrukcje warunkowe, pętle, metody, funkcje itp.

Nabywanie przez studenta umiejętności tworzenia aplikacji w języku C# z wykorzystaniem konsoli.

Samodzielne radzenie sobie w sytuacji trudnej poprzez analizę problemu oraz pomoc w jego rozwiązaniu przy pomocy dostępnych źródeł informacji.

**Metody dydaktyczne:**

metody dydaktyczne:				metody oceniania:	
MP1	wykład informacyjny			MO1	kolokwium pisemne
MP2	objaśnienie			MO2	aktywność w trakcie zajęć
MP3	praca ze źródłem elektronicznym				
MC1	ćwiczenie praktyczne				
MS1	metoda sytuacyjna				

**Wykład**

W1	Tworzenie graficznego interfejsu użytkownika w języku C#(okno aplikacji,
W2	Programowe łączenie z relacyjną bazą danych i wykonywanie zapytań do bazy danych w języku C# (wybrana technologia bazodanowa np.: MySQL, PostgreSQL, SQLite itp.).
W3	Tworzenie aplikacji internetowych działających po stronie serwera w języku C#.
W4	Tworzenie aplikacji działających w architekturze klient – serwer w języku C#.
W5	Wprowadzenie do platformy .NET i języka C#. Porównanie technologii .NET i Java. Obsługa środowiska Microsoft Visual Studio .NET.
W6	Podstawowe typy danych, wyrażenia, instrukcje w języku C#. Tablice jedno i wielowymiarowe.
W7	Programowanie obiektowe w C#: tworzenie klas i obiektów, pola i metody, hermetyzacja, kompozycja, dziedziczenie, abstrakcja, polimorfizm
W8	Operacje na strumieniach. Obsługa wyjątków w języku C#.
W9	Obsługa kontenerów w języku C#.
W10	Tworzenie graficznego interfejsu użytkownika przy użyciu standardowych komponentów .NET.
W11	Obsługa zdarzeń myszy i klawiatury, programowanie ruchomej grafiki 2D w języku C#

**Ćwiczenia**

C1	Tworzenie GUI w języku C#
C2	Łączenie się z relacyjną bazą danych w języku C#
C3	Tworzenie
C4	Tworzenie aplikacji działających w architekturze klient – serwer w języku C#
C5	Tworzenie własnego projektu wraz z dokumentacją w postaci dokletu w języku C#.
C6	Wprowadzenie do środowiska Visual Studio i SharpDevelop.
C7	Typy danych, wyrażenia warunkowe i instrukcje sterujące w języku C#.
C8	Tworzenie klas, obiektów i struktur w języku C#.
C9	Hermetyzacja, dziedziczenie i kompozycja w języku C#.
C10	Polimorfizm i abstrakcja (klasy abstrakcyjne i interfejsy) w języku C#.
C11	Kolekcje w języku C#.
C12	Obsługa wyjątków w języku
C13	Operacje na strumieniach w języku C#.
C14	Tworzenie GUI i obsługa zdarzeń w języku C#.

**Projekt**

P1	Omówienie najpopularniejszych algorytmów sortujących
P2	Każdy uczeń tworzy program w oparciu o wybrany algorytm sortujący wraz z dokumentacją
P3	Prezentacja poszczególnych projektów wraz z ich omówieniem

**Literatura podstawowa**

1	C. S. Horstmann: Core Java 2. [T. 1], Podstawy, Gliwice, Helion, 2003
2	C. S. Horstmann: Core Java 2. [T. 2], Techniki zaawansowane, Gliwice, Helion,
3	A. Troelsen: Język C# 2010 i platforma .NET 4, Warszawa, PWN, 2014
4	M. Lis: C# : praktyczny kurs : poznaj tajniki programowania w C#, Gliwice,
5	Eckel B., Thinking in Java. Wydanie 4, edycja polska, Helion, Gliwice, 2006.

**Literatura uzupełniająca**

1	Schildt H., Java. Kompedium programisty, Helion, Gliwice, 2005.
2	Michelsen K., Język C#. Szkoła programowania, Helion, Gliwice, 2007.

3	Liberty J., Programowanie C#, Helion, Gliwice, 2006.
4	Kniat J., Programowanie w języku C++, NAKOM, Poznań, 1999.

<b>Warunki zaliczenia</b> Warunkiem zaliczenia jest, uzyskanie oceny, co najmniej dostatecznej ze wszystkich kolokwiów oraz zaliczenie egzaminu z wiedzy teoretycznej. Ocena końcowa zależy od liczby uzyskanych punktów ze wszystkich składowych oceniania i wynosi: do 50% - ocena niedostateczna, 51% - 60% - ocena dostateczna, 61% - 70% - ocena dostateczna plus, 71% - 80% - ocena dobra 81% - 90% - ocena dobra plus, 91% - 100% - ocena bardzo dobra.
<b>Przykłady pytań zaliczeniowych</b> Co to jest programowanie obiektowe? Co to jest dziedziczenie ? Co to jest polimorfizm? Co to jest hermetyzacja?

**Obciążenie pracą studenta**  
*Studia stacjonarne/niestacjonarne*

Forma pracy studenta	Wykład		Ćwiczenia		Projekt		Suma	
Zajęcia z bezpośrednim udziałem nauczyciela	15 g	8 g	45 g	24 g	15 g	8 g	75 g	40 g
Zapoznanie się z literaturą przedmiotu	1 g	5 g	1 g	5 g			2 g	10 g
Przygotowanie się do zajęć			30 g	30 g			30 g	30 g
Przygotowanie się do kolokwiów								
Realizacja zadanych ćwiczeń i zadań			40 g	40 g			40 g	40 g
Przygotowanie sprawozdania z ćwiczeń								
Przygotowanie projektu / pracy			3 g	30 g			3 g	30 g
Przygotowanie się i udział w egzaminie								
	16 g	13 g	119 g	129 g	15 g	8 g	150 g	150 g

Efekty uczenia się	KEK	Treści kształcenia	Metody dydaktyczne	M. oceniania
rozumie i opisuje podstawowe cechy systemu opartego o programowanie obiektowe.	K_U03	W7-W8 C3-C6 P1-P3	MP1-MP3	MO1
Student ma uporządkowaną i podbudowaną teoretycznie wiedzę w zakresie podstawowych algorytmów i ich analizy, technik projektowania algorytmów, abstrakcyjnych struktur danych i ich implementacji, problemów obliczeniowo trudnych	K_U04	W1-W8 C1-C8 P1-P3	MP1-MP3	MO1
Student ma uporządkowaną i podbudowaną teoretycznie wiedzę w zakresie podst. konstrukcji programistycznych, implementacji algorytmów, paradygmatów i stylów programowania, metod weryfikacji poprawności programów, języków formalnych, kompilatorów, platform programistycznych	K_U05	W1-W8 C1-C8 P1-P3	MP1-MP3	MO1
Korzysta z różnych źródeł w celu porównania różnych rozwiązań i wyboru najkorzystniejszego.	K_U01	W1-W12 C3-C13 P1-P3	MP1-MP3	MO2
potrafi dobrać właściwe polecenia oraz narzędzia do realizacji typowych zadań.	K_U04	W1-W12 C3-C13 P1-P3	MP1-MP3	MO2
potrafi wykorzystywać posiadaną wiedzę z zakresu programowania poprzez umiejętność formułowania i rozwiązywania zastosowań struktur algorytmicznych w rozwiązywaniu problemów praktycznych	K_U05	W1-W12 C3-C13 P1-P3	MP1-MP3	MO2
Student potrafi posłużyć się środowiskami i platformami programistycznymi do pisania, wykonywania i testowania prostych programów kodowanych w językach programowania imperatywnego, obiektowego i deklaratywnego.	K_U10	W1-W12 C3-C13 P1-P3	MP1-MP3	MO2
jest gotów do rozwiązywania problemów praktycznych, korzystając samodzielnie z dostarczonych materiałów lub korzystając z konsultacji wykładowcy oraz przygotowuje się do właściwej realizacji zadań podczas testów	K_K02	W1-W12 C1-C15 P1-P3	MP1-MP3	MO2
jest gotów do dzielenia się swoją wiedzą i umiejętnościami na rzecz środowiska grupy studenckiej, publikując na forum dyskusyjnym wpisy z rozwiązaniami zadań problemowych	K_K03	W1-W12 C1-C15 P1-P3	MP1-MP3	MO2

docenia zalety i dostrzega wady wolnego oprogramowania oraz uzmysławia sobie społeczną rolę programistów systemu Linux w rozpowszechnianiu wykorzystania komputerów przez użytkowników	K_K05	W1-W12 C1-C15 P1-P3	MP1-MP3	MO2
docenia zalety i dostrzega wady wolnego oprogramowania oraz uzmysławia sobie społeczną rolę programistów systemu Linux w rozpowszechnianiu wykorzystania komputerów przez użytkowników	U_K07	W1-W12 C1-C15 P1-P3	MP1-MP3	MO2