



**KAPITAŁ LUDZKI**  
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez  
Unię Europejską w ramach  
Europejskiego Funduszu  
Społecznego

**UNIA EUROPEJSKA**  
EUROPEJSKI  
FUNDUSZ SPOŁECZNY



<b>Nazwa przedmiotu</b>		<b>Kod ECTS</b>	
Wstęp do programowania		11.3.0722	
<b>Nazwa jednostki prowadzącej przedmiot</b>			
Faculty of Mathematics, Physics and Informatics			
<b>Studia</b>			
<b>wydział</b>	<b>kierunek</b>	<b>poziom</b>	<b>pierwszego stopnia</b>
Wydział Matematyki, Fizyki i Informatyki	Informatyka	forma	stacjonarne
		moduł	wszystkie
		specjalnościowy specjalizacja	wszystkie
<b>Nazwisko osoby prowadzącej (osób prowadzących)</b>			
dr Janusz Dybizbański; prof. UG, dr hab. Marcin Wieśniak; dr inż. Jerzy Skurczyński; dr Magdalena Godlewska; mgr Omer Sakarya; mgr Gabriela Łuczyńska; dr Mikołaj Czechlewski; mgr Mateusz Miotk; mgr inż. Anna Nenca; mgr inż. Anna Nenca; dr Paweł Pączkowski; mgr Radosław Ziemann; dr Andrzej Borzyszkowski; dr Piotr Arłukowicz; dr Karol Horodecki; dr Marcin Ciecholewski; mgr Maciej Dziemiańczuk; mgr Adam Kostulak; prof. UG, dr hab. Tomasz Dzido			
<b>Formy zajęć, sposób ich realizacji i przypisana im liczba godzin</b>		<b>Liczba punktów ECTS</b>	
<b>Formy zajęć</b>		8 Przedmiot w wymiarze 30h wykładu, 30h ćw. aud. i 30h ćw. laboratoryjnych + praca własna studenta	
Wykład, Ćw. audytoryjne, Ćw. laboratoryjne			
<b>Sposób realizacji zajęć</b>			
zajęcia w sali dydaktycznej			
<b>Liczba godzin</b>			
Ćw. laboratoryjne: 30 godz., Ćw. audytoryjne: 30 godz., Wykład: 30 godz.			
<b>Cykl dydaktyczny</b>			
2016/2017 zimowy			
<b>Status przedmiotu</b>		<b>Język wykładowy</b>	
obowiązkowy		polski	
<b>Metody dydaktyczne</b>		<b>Forma i sposób zaliczenia oraz podstawowe kryteria oceny lub wymagania egzaminacyjne</b>	
<ul style="list-style-type: none"> <li>- Rozwiązywanie zadań</li> <li>- Wykład z prezentacją multimedialną</li> <li>- ćwiczenia laboratoryjne -- sporządzanie i uruchamianie programów komputerowych</li> </ul>		<b>Sposób zaliczenia</b>	
		<ul style="list-style-type: none"> <li>- Zaliczenie na ocenę</li> <li>- Egzamin</li> </ul>	
		<b>Formy zaliczenia</b>	
		<ul style="list-style-type: none"> <li>- egzamin pisemny z pytaniami (zadaniami) otwartymi</li> <li>- ustalenie oceny zaliczeniowej na podstawie ocen cząstkowych otrzymywanych w trakcie trwania semestru</li> <li>- kolokwium</li> <li>- egzamin pisemny (dłuższa wypowiedź pisemna / rozwiązanie problemu)</li> </ul>	
		<b>Podstawowe kryteria oceny</b>	
		Przedmiot kończy się egzaminem z pytaniami otwartymi. Warunkiem przystąpienia do egzaminu jest uzyskanie pozytywnych ocen zarówno z ćwiczeń laboratoryjnych jak i ćwiczeń audytoryjnych.	
<b>Sposób weryfikacji założonych efektów kształcenia</b>			

zakładany efekt kształcenia	egzamin	kolokwium	projekt	referat	raport	aktywność w dyskusji	obserwacja i ocena postawy i umiejętności studenta
Wiedza							
K_W02	X	X	X			X	X
K_W03	X	X	X			X	X
K_W04	X	X	X			X	X
Umiejętności							
K_U01	X	X	X			X	X
K_U03	X	X	X			X	X
K_U05			X			X	X
K_U06	X	X	X			X	X
K_U08	X	X	X			X	X
Kompetencje							
K_K02						X	X

## Określenie przedmiotów wprowadzających wraz z wymogami wstępnymi

### A. Wymagania formalne

brak

### B. Wymagania wstępne

brak

## Cele kształcenia

Celem przedmiotu jest poznanie podstawowych technik programistycznych oraz zdobycie umiejętności projektowania, analizy i implementacji podstawowych algorytmów.

## Treści programowe

- Algorytm a program. Proste programy w różnych sposobach zapisu (opis słowny, schemat blokowy, instrukcje języka programowania). Ręczna symulacja działania algorytmu.
- Schemat przetwarzania programu w języku wysokiego poziomu - etapy kompilacji, konsolidacji i wykonania.
- Deklaracje zmiennych. Proste typy danych oraz typy strukturalne: tablica, rekord. Zakres deklaracji i widoczności zmiennych, zmienne globalne.
- Instrukcje sterujące. Pojęcie poprawności częściowej i całkowitej programu. Dowodzenie poprawności metodą niezmienników. Zagnieżdżone pętle.
- Procedury i funkcje. Sposoby przekazywania parametrów do funkcji.
- Mechanizm rekursji i jego wykorzystanie. Wyrażanie pętli przez rekursję i przykład wyrażenia rekursji pętli. Dowodzenie poprawności funkcji rekurencyjnych za pomocą indukcji matematycznej.
- Szacowanie liczby operacji wykonywanych przez algorytm.
- Reprezentacja liczb całkowitych i zmiennoprzecinkowych w komputerze oraz błędy nimi spowodowane (przekroczenie zakresu i błędy zaokrągleń).
- Wykorzystanie wskaźników i zarządzanie pamięcią na przykładzie prostych struktur danych.
- Informacja o teoretycznych ograniczeniach obliczeniowych.

## Wykaz literatury

- [1] Griffiths David, Griffiths Dawn. *C. Rusz głową!* Wydawnictwo Helion.  
 [2] Szepietowski A. *Podstawy informatyki.* Wydawnictwo Uniwersytetu Gdańskiego, Gdańsk 2000.  
 [3] Kernighan B.W., Ritchie D.M. *Język ANSI C.* Wydawnictwa Naukowo-Techniczne, Warszawa 2003.

## Efekty kształcenia (obszarowe i kierunkowe)

K\_W02 ma uporządkowaną, podbudowaną teoretycznie wiedzę ogólną w zakresie programowania, algorytmów i złożoności, architektury systemów komputerowych, systemów operacyjnych, technologii sieciowych, języków i paradygmatów programowania, baz danych, inżynierii

## Wiedza

Student:

- zna podstawowe narzędzia programistyczne
- zna elementy programów komputerowych: zmienne, instrukcje proste i sterujące, funkcje
- ma podstawy wiedzy o sposobach analizy poprawności i złożoności algorytmów

<p>oprogramowania, języków formalnych, K_W03 zna podstawowe metody projektowania, analizowania i programowania algorytmów, K_W04 zna podstawowe konstrukcje programistyczne oraz struktury danych, K_U01 potrafi zastosować wiedzę matematyczną do formułowania, analizowania i rozwiązywania prostych zadań związanych z informatyką, K_U03 potrafi pracować indywidualnie i w zespole informatyków, w tym także potrafi zarządzać swoim czasem oraz podejmować zobowiązania i dotrzymywać terminów, K_U05 potrafi pisać, uruchamiać i testować programy w wybranym środowisku programistycznym K_U06 projektuje, analizuje pod kątem poprawności i złożoności obliczeniowej oraz programuje algorytmy; wykorzystuje podstawowe techniki algorytmiczne i struktur danych K_U08 posługuje się przyjętymi formatami reprezentacji różnego rodzaju danych stosownie do sytuacji K_K02 potrafi precyzyjnie formułować pytania, służące pogłębieniu własnego zrozumienia danego tematu lub odnalezieniu brakujących elementów rozumowania</p>	<ul style="list-style-type: none"> <li>- zna typowe algorytmów obliczania wyszukiwania i porządkowania</li> <li>- zna mechanizm rekursji</li> <li>- zna sposoby reprezentacji liczb w komputerze</li> <li>- zna podstawy technik programistycznych: dziel i rządź, DP</li> <li>- zna podstawowe struktury danych: stos, kolejka</li> </ul>
	<p><b>Umiejętności</b></p> <p>Student:</p> <ul style="list-style-type: none"> <li>- umie przeprowadzić ręczną symulacja działania prostego programu z pętlami lub rekursją</li> <li>- umie zaprojektować i zaimplementować program z pętlami (również zagnieżdżonymi) oraz funkcjami</li> <li>- umie uzasadnić, że napisany program działa zgodnie z (nieformalnymi) wymaganiami</li> <li>- potrafi ocenić liczbę operacji, wykonywanych przez program z pętlami</li> <li>- umie zaprojektować proste funkcje rekurencyjne</li> <li>- potrafi zapisywać i odczytywać liczby jako ciągu bitów, zarówno w przypadku stało- jak i zmienna-pozycyjnym</li> </ul>
	<p><b>Kompetencje społeczne (postawy)</b></p> <p>Student:</p> <ul style="list-style-type: none"> <li>- potrafi precyzyjnie formułować pytania, służące pogłębieniu własnego zrozumienia danego tematu lub odnalezieniu brakujących elementów rozumowania</li> </ul>
<p><b>Kontakt</b></p> <p>jdybiz@inf.ug.edu.pl</p>	