



## KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Systemy informatyczne [S1ETI2>SI]

### Przedmiot

Kierunek studiów

Edukacja techniczno-informatyczna

Rok/Semestr

2/3

Studia w zakresie (specjalność)

–

Profil studiów

ogólnoakademicki

Poziom studiów

pierwszego stopnia

Język oferowanego przedmiotu

polski

Forma studiów

stacjonarne

Wymagalność

obligatoryjny

### Liczba godzin

Wykład

15

Laboratorium

30

Inne

0

Ćwiczenia

0

Projekty/seminaria

0

### Liczba punktów ECTS

4,00

### Koordynatorzy

dr hab. inż. Małgorzata Jankowska  
malgorzata.jankowska@put.poznan.pl

### Wykładowcy

### Wymagania wstępne

Podstawowa wiedza z informatyki. Umiejętność biegłego programowania w języku C/C++, algorytmizacji zadań oraz logicznego i abstrakcyjnego myślenia. Zrozumienie potrzeby tworzenia systemów informatycznych w celu zwiększenia wydajności pracy, obliczeń, wizualizacji wyników oraz prezentacji informacji zgromadzonych w bazach danych.

### Cel przedmiotu

Głównym celem przedmiotu jest przekazanie wiedzy dotyczącej zaawansowanych elementów języka C/C++, których znajomość umożliwi tworzenie autorskich, złożonych systemów informatycznych. Ponadto zakłada się wprowadzenie do programowania w języku Java.

### Przedmiotowe efekty uczenia się

Wiedza:

Wiedza:

zna i rozumie zaawansowane elementy języka C/C++, wykorzystywane do tworzenia aplikacji informatycznych, w tym programowanie obiektowe, przeciążanie operatorów, obsługę wyjątków, dynamiczne struktury danych, przestrzenie nazw.

ma wiedzę niezbędną do tworzenia aplikacji w języku Java.

#### Umiejętności:

ma umiejętność tworzenia programów komputerowych z wykorzystaniem zaawansowanych elementów języka programowania C/C++ oraz tworzenia prostych programów w języku Java.

posiada umiejętność pozyskiwania informacji z literatury, baz danych oraz innych dostępnych źródeł wiedzy oraz umiejętność pracy indywidualnej i w zespole.

#### Kompetencje społeczne:

potrafi dzielić się zdobytą wiedzą informatyczną z innymi w sposób zrozumiały.

widzi znaczenie informatyzacji różnych dziedzin życia w odniesieniu do człowieka i społeczeństwa.

rozumie potrzebę ciągłego dokształcania się w celu podnoszenia kompetencji zawodowych, zdobywania aktualnej wiedzy z zakresu ogólnie pojętej informatyki oraz języków programowania (np. poprzez czytanie czasopism informatycznych, uczestnictwie w kursach i studiach podyplomowych).

wykazać się odpowiedzialnością za powierzone mu zadania, np. samodzielny projekt programistyczny.

potrafi pracować nad wyznaczonym zadaniem samodzielnie oraz w zespole przyjmując w nim różne role.

### Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

Pisemne kolokwia obejmujące materiał wykazany w treściach programowych. Ocena końcowa uzależniona od procentowej liczby punktów:

3 50.1%-70.0%

4 70.1%-90.0%

5 od 90.1%

Zadania projektowe wymagane w formie programów i oceniane pod koniec zajęć laboratoryjnych oraz na kolokwiach zaliczeniowych. Ocena końcowa uzależniona od procentowej liczby punktów:

3 50.1%-70.0%

4 70.1%-90.0%

5 od 90.1%

### Treści programowe

Tematyka modułu obejmuje zaawansowane elementy języka C++ wykorzystywane do tworzenia funkcji oraz ponadto zagadnienia dotyczące programowania obiektowego w języku C++. Kolejna grupa zagadnień dotyczy wprowadzenia do programowania w języku Java. W ramach treści programowych zakłada się omówienie podstaw języka Java oraz elementów programowania obiektowego w tym języku.

### Tematyka zajęć

Wykłady:

1. Programowanie obiektowe:

- klasy, obiekty klas,
- przeciążanie operatorów,
- obsługa wyjątków,
- dziedziczenie,
- polimorfizm.

2. Wprowadzenie do języka programowania Java:

- podobieństwa i różnice języków C++ i Java,
- zalety języka Java i możliwości zastosowania tego języka do tworzenia aplikacji,
- programowanie obiektowe w języku Java.

Laboratoria:

1. Zaawansowane elementy języka C++ wykorzystywane do tworzenia funkcji:

- tablice statyczne i dynamiczne, jedno- i dwuwymiarowe jako argumenty funkcji,
- rekurencja,
- modyfikator const,
- wskaźniki i referencje jako wynik funkcji,
- przeciążanie funkcji,
- szablony funkcji.

2. Programowanie obiektowe:

- klasy, obiekty klas,
- przeciążanie operatorów,
- obsługa wyjątków,
- dziedziczenie,
- polimorfizm.

### 3. Wprowadzenie do języka programowania Java:

- podobieństwa i różnice języków C++ i Java,
- zalety języka Java i możliwości zastosowania tego języka do tworzenia aplikacji,
- programowanie obiektowe w języku Java.

## Metody dydaktyczne

### Wykłady:

Prezentacja multimedialna ilustrowana przykładowymi programami napisanymi w językach C/C++ i Java.

### Laboratoria:

Tworzenie programów komputerowych w językach C/C++ i Java.

## Literatura

### Podstawowa:

1. H. M. Deitel, P. J. Deitel, Arkana C++ Programowanie, Wydawnictwo RM, Warszawa 1998.
2. S. Prata, Szkoła Programowania. Język C++, Wydawnictwo Helion, Gliwice 2006.
3. A. Zalewski, Programowanie w językach C i C++ z wykorzystaniem pakietu Borland C++, Wydawnictwo Nakom, Poznań 1996.
4. J. Grębosz, Symfonia C++. Programowanie w języku C++ orientowane obiektowo, Tom 1,2,3, Oficyna Kallimach, Kraków 1999.
5. H. Schildt, Java. Kompendium programisty, Wydanie XII, Helion, Gliwice 2023.
6. K. Sierra, B. Bates, T. Gee, Java. Rusz głową! Przewodnik po praktycznym programowaniu w Javie, Wydanie III, Helion, Gliwice 2023.

### Uzupełniająca:

1. D. E. Knuth, Sztuka programowania, tom1. Algorytmy podstawowe, Wydawnictwa Naukowo-Techniczne, Warszawa 2002.
2. N. Wirth, Algorytmy + struktury danych = programy, Wydawnictwa Naukowo-Techniczne, Warszawa 2004.

## Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	100	4,00
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	45	2,00
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych/ćwiczeń, przygotowanie do kolokwium/egzaminu, wykonanie projektu)	55	2,00