**Nazwa przedmiotu:**

Zasady programowania strukturalnego II

**Koordynator przedmiotu:**

dr inż. Maciej Przybylski

**Status przedmiotu:**

Obowiązkowy

**Poziom kształcenia:**

Studia I stopnia

**Program:**

Automatyka Robotyka i Informatyka Przemysłowa

**Grupa przedmiotów:**

Obowiązkowe

**Kod przedmiotu:**

ZAPIIst

**Semestr nominalny:**

2 / rok ak. 2018/2019

**Liczba punktów ECTS:**

3

**Liczba godzin pracy studenta związanych z osiągnięciem efektów uczenia się:**

1) Liczba godzin kontaktowych: 20 godz., w tym:
• projektowanie w pracowni komputerowej - 15 godz.
• konsultacje - 5 godz.
Praca własna studenta – 60 godz., w tym:
• studia literaturowe - 10 godz.
• opracowanie projektu poza pracownią - 35 godz.
• opracowanie dokumentacji projektu - 15 godz.
Razem: 80 godz. (3 ECTS)

**Liczba punktów ECTS na zajęciach wymagających bezpośredniego udziału nauczycieli akademickich:**

1 punkt ECTS - liczba godzin kontaktowych: 20 godz., w tym:
• projektowanie w pracowni komputerowej - 15 godz.
• konsultacje - 5 godz.

**Język prowadzenia zajęć:**

polski

**Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym:**

3 punkty ECTS – 75 godz., w tym:
• obecność na zajęciach - 15 godz.
• projektowania poza pracownią - 45 godz.
• dokumentowanie projektu - 15 godz.

**Formy zajęć i ich wymiar w semestrze:**

|  |  |
| --- | --- |
| Wykład: | 0h |
| Ćwiczenia: | 0h |
| Laboratorium: | 0h |
| Projekt: | 15h |
| Lekcje komputerowe: | 0h |

**Wymagania wstępne:**

Zaliczony (wskazane) przedmiot: Zasady programowania strukturalnego I.

**Limit liczby studentów:**

max. 15 studentów

**Cel przedmiotu:**

Nauka logicznego, algorytmicznego myślenia przy rozwiązywaniu problemów inżynierskich.
Praktyczne wykorzystanie w programowaniu strukturalnym poznanych złożonych dynamicznych struktur danych: list, drzew i grafów.
Zdobycie umiejętności tworzenia złożonych programów wraz z tworzeniem bibliotek funkcji definiowanych we własnych modułach.
Wprowadzenie do programowania zorientowanego obiektowo, w tym wizualno-obiektowego.

**Treści kształcenia:**

Indywidualne projekty są realizowane w języku C/C++. W tworzonej aplikacji wymagane jest: 1) wprowadzanie i/lub zapisywanie danych do programu z wykorzystaniem typu plikowego, 2) struktura danych do projektu musi też wykorzystywać: tablice dynamiczne, listy, drzewa lub grafy, 3) aplikacja musi mieć budowę modułową i funkcje programu muszą być umieszczane w utworzonych dla niej bibliotekach. Aplikacja musi być udokumentowana i testowana. Wprowadzenie do programowania wizualno-obiektowego obejmuje: definicję typu, definicje obiektów i metod, dziedziczenie i poliformizm, typowe komponenty wizualne oraz podstawy pisania aplikacji wizualnych.

**Metody oceny:**

ZALICZENIE Ocena zaliczająca przedmiot jest ustalana na podstawie projektu indywidualnie opracowanego w trakcie semestru. Ocena projektu składa się z ocen cząstkowych za: dokumentację wstępną, prototyp, ostatecznie wykonanie aplikacji w wersji konsolowej i okienkowej. Wymagana jest również osobista prezentacja projektu oraz umiejętność wyjaśnienia szczegółów przedstawionej aplikacji.

**Egzamin:**

nie

**Literatura:**

1) P. Wnuk, B. Putz: Informatyka 2 - Programowanie. Wersja w języku C/C++. Podręcznik internetowy, OKNO PW, 2004-2010.
2) B. Putz, A. Putz jr, P. Wnuk : Algorytmy i struktury danych. Podręcznik internetowy, OKNO PW, 2006-2010.
3) S. Prata: Język C++. Wydanie V, Helion 2006.
4) J. Grębosz: Symfonia C++ standard. Tom I. Edition 2000 Kraków, 2005-2008.
5) B. Eckel: Thinking in C++. Edycja polska. Helion 2002.
6) N. Wirth: Algorytmy+struktury danych=programy. WNT 2004.
7) P. Wróblewski: Algorytmy, struktury danych i techniki programowania. Wyd. IV, Helion 2010.
8) T.H. Cormen et al.: Wprowadzenie do algorytmów. WNT 2007, PWN 2012 (nowe wydanie).

**Witryna www przedmiotu:**

http://iair.mchtr.pw.edu.pl/studenci/

**Uwagi:**

Przedmiot daje gruntowne podstawy do programowania, umożliwiając tworzenie własnych aplikacji do rozwiązywania różnych problemów technicznych.

## Efekty przedmiotowe

### Profil ogólnoakademicki - wiedza

**Efekt ZAPIIst\_W01:**

Posiada praktyczną wiedzę w zakresie algorytmizacji problemów technicznych

Weryfikacja:

Zaliczenie projektu

**Powiązane efekty kierunkowe:** K\_W04, K\_W05

**Powiązane efekty obszarowe:** T1A\_W02, T1A\_W02, T1A\_W04

**Efekt ZAPIIst\_W02:**

Ma wiedzę w zakresie programowania strukturalnego

Weryfikacja:

Zaliczenie projektu

**Powiązane efekty kierunkowe:** K\_W04, K\_W05

**Powiązane efekty obszarowe:** T1A\_W02, T1A\_W02, T1A\_W04

### Profil ogólnoakademicki - umiejętności

**Efekt ZAPIIst\_U01:**

Ma praktyczną umiejętność tworzenia złożonych programów

Weryfikacja:

Zaliczenie projektu

**Powiązane efekty kierunkowe:** K\_U08, K\_U09

**Powiązane efekty obszarowe:** T1A\_U09, T1A\_U09

### Profil ogólnoakademicki - kompetencje społeczne

**Efekt ZAPIIst\_K01:**

Rozumie potrzebę ciągłego kształcenia

Weryfikacja:

Zaliczenie

**Powiązane efekty kierunkowe:** K\_K01

**Powiązane efekty obszarowe:** T1A\_K01