**Nazwa przedmiotu:**

Informatyka II

**Koordynator przedmiotu:**

Tomasz Łukasiak, dr inż., Tomasz Sokół, dr inż.

**Status przedmiotu:**

Obowiązkowy

**Poziom kształcenia:**

Studia I stopnia

**Program:**

Budownictwo

**Grupa przedmiotów:**

Obowiązkowe

**Kod przedmiotu:**

1080-BU000-IZP-0307

**Semestr nominalny:**

4 / rok ak. 2019/2020

**Liczba punktów ECTS:**

2

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

Razem 50 godz. = 2 ECTS: 20 godz. w laboratorium komputerowym, 30 godz. pracy samodzielnej w domu w celu ugruntowania zdobytej na zajęciach wiedzy.
Praca domowa pozwala na połączenie wszystkich poznanych struktur programowania w jednym programie oraz umożliwia modyfikowanie wprowadzonego kodu w celu eliminacji błędów oraz jego ulepszenia pod względem efektywności numerycznej.

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

Razem 20 godz. zajęć laboratoryjnych = 1 ECTS

**Język prowadzenia zajęć:**

polski

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

Razem 50 godz. = 2 ECTS: ćwiczenia laboratoryjne w pracowni komputerowej 20 godz., praca własna związana z ugruntowaniem wiedzy przekazanej na ćwiczeniach oraz pisanie własnych wersji programów 30 godz.

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

|  |  |
| --- | --- |
| Wykład:  | 0h |
| Ćwiczenia:  | 20h |
| Laboratorium:  | 0h |
| Projekt:  | 0h |
| Lekcje komputerowe:  | 0h |

**Wymagania wstępne:**

Znajomość obsługi systemu Windows oraz pakietu Office. Podstawowa znajomość programu Excel. Umiejętność tworzenia folderów, zarządzania plikami (kopiowanie , zmiana nazwy i rozszerzenia, wyszukiwanie, edycja edytorem ASCII) za pomocą dowolnego menadżera plików. Znajomość zasad określania ścieżki dostępu (względnej i bezwzględnej ) do pliku w systemie Windows.

**Limit liczby studentów:**

brak

**Cel przedmiotu:**

Celem przedmiotu jest nauczenie podstaw algorytmizacji problemu, automatyzacji obliczeń z zastosowaniem makropoleceń oraz samodzielnego programowania. Wyjaśniane są zasady poprawnego formułowania problemu, podziału na czynności elementarne oraz podstawy programowania. Nacisk położony jest na naukę podstawowych instrukcji sterujących, typów danych, organizację operacji wejścia/wyjścia oraz proste programowanie proceduralne. W trakcie zajęć studenci poznają nowoczesne środowiska programistyczne oraz podstawowe możliwości ich konfiguracji. Uczą się programować w wybranym języku proste algorytmy oraz samodzielnie testować poprawność napisanych programów m.in. za pomocą wbudowanych debugerów. Zakłada się, że po ukończonym kursie, student będzie samodzielnie potrafił zalgorytmizować problem oraz napisać program realizujący poprawnie ten algorytm. Poznanie podstawowych zasad programowania, terminologii oraz metod pozwoli w przyszłości na efektywniejsze użytkowanie profesjonalnych programów zawodowych takich jak np. AutoCad, Abaqus, AnSys itp., które umożliwiają pisanie i wykorzystanie w praktyce inżynierskiej własnych makr, procedur oraz programów

**Treści kształcenia:**

1. Zapoznanie się z graficznym interfejsem użytkownika środowiska programistycznego – konfiguracja, edycja, kompilacja i uruchamiania programów.
2. Semantyka typów prostych i złożonych, deklaracje i definicje zmiennych, operatory arytmetyczne i logiczne, wyrażenia warunkowe, priorytet i łączność operatorów, wyrażenia logiczne, instrukcje sterujące, pętle indeksowe i warunkowe, podstawowe operacje wejścia i wyjścia, podstawowe struktury danych: tablice jedno- i wielowymiarowe, rekordy, funkcje i procedury, zakres ważności zmiennych, przekazywanie argumentów.
3. Paradygmaty programowania proceduralnego: podział zadania na rozłączne etapy i ich realizacja w podprogramach.
4. Uruchamianie samodzielnie napisanych programów poza środowiskiem programistycznym – bezpośrednio w systemie komputerowym. Informacje na temat programowej obsługi błędów oraz programowania obiektowego.

**Metody oceny:**

Na każdych ćwiczeniach, studenta obowiązuje znajomość teoretyczna i praktyczna przerobionego materiału.
Przygotowanie studenta do zajęć oceniana jest przez prowadzącego ćwiczenia w trakcie ćwiczeń.
Ocena i zaliczenie przedmiotu zależą od liczby punktów otrzymanych ze sprawdzianu lub sprawdzianu poprawkowego. Oba sprawdziany trwają 1 godzinę (60 minut) i polegają na samodzielnym napisaniu i uruchomieniu w środowisku Excel kilku (3-5) modułów (makr, procedur, programów) ocenianych punktowo. Ocenę pozytywną ze sprawdzianu otrzymuje się po zdobyciu co najmniej 50% + 1 pkt. punktów.
Prowadzący może zwiększyć tę ocenę biorąc pod uwagę aktywność studenta oraz jego przygotowanie do zajęć.
Na sprawdzianie poprawkowym nie można otrzymać oceny wyższej od oceny dostatecznej. Termin sprawdzianu poprawkowego (w czasie sesji i tylko dla osób, które nie zdobyły dostatecznej do zaliczenia liczby punktów) ustalany jest po zakończeniu zajęć.
Możliwe jest indywidualne zaliczenie przedmiotu na podstawie samodzielnie napisanego programu w dowolnym.

**Egzamin:**

nie

**Literatura:**

[1] O programowaniu inaczej, R.L.Baber, WNT 1989;
[2] Algorytmy + Struktury Danych = Programowanie, N. Wirth, WNT 1989;
[3] Algorytmy, struktury danych i techniki programowania. Piotr Wróblewski,Wydanie IV – Helion 2009;
[4] Siedem języków w siedem tygodni. Praktyczny przewodnik nauki języków programowania - , Bruce A. Tate, Helion 2011;
[5] Algorytmy. Wydanie IV -,Robert Sedgewick, Kevin Wayne, Helion 2011;
[6] Czysty kod. Podręcznik dobrego programisty, Robert C. Martin, Helion 2009;
[7] Excel 2010 PL. Programowanie w VBA. Vademecum Walkenbacha, J.Walkenbach, Helion 2011;
[8] Getting Started with VBA in Excel 2010, msdn.microsoft.com 2011;
[9] Excel 2007 PL. Programowanie w VBA. J.Green, S. Bullen, R. Bovey, M.Alexander, Helion 2008.

**Witryna www przedmiotu:**

http://wektor.il.pw.edu.pl/~iap/

**Uwagi:**

## Efekty przedmiotowe

### Profil ogólnoakademicki - wiedza

**Efekt W1:**

Zna i rozumie pojęcia z zakresu semantyki i algorytmizacji formułowane w wybranym środowisku programowania. Zna podstawowe parametry konfiguracji środowiska programowania.

Weryfikacja:

Sprawdzian z umiejętności algorytmizacji i programowania.

**Powiązane efekty kierunkowe:** K1\_W09

**Powiązane efekty obszarowe:** T1A\_W01, T1A\_W02, T1A\_W05, T1A\_W07

### Profil ogólnoakademicki - umiejętności

**Efekt U1:**

Zna podstawowe algorytmy numeryczne. Potrafi zastosować i połączyć ze sobą odpowiednie algorytmy w celu rozwiązania złożonego problemu numerycznego.

Weryfikacja:

Sprawdzian z umiejętności programowania proceduralnego.

**Powiązane efekty kierunkowe:** K1\_U06

**Powiązane efekty obszarowe:** T1A\_U01, T1A\_U07, T1A\_U08, T1A\_U09, T1A\_U15

### Profil ogólnoakademicki - kompetencje społeczne

**Efekt K1:**

Samodzielnie uzupełnia i poszerza wiedzę.

Weryfikacja:

Zadania do samodzielnego rozwiązania poza zajęciami.

**Powiązane efekty kierunkowe:** K1\_K01, K1\_K03

**Powiązane efekty obszarowe:** T1A\_K03, T1A\_K01, T1A\_K05, T1A\_K06