**Nazwa przedmiotu:**

Metody translacji

**Koordynator przedmiotu:**

Dr inż. Jan Bródka

**Status przedmiotu:**

Obowiązkowy

**Poziom kształcenia:**

Studia I stopnia

**Program:**

Informatyka

**Grupa przedmiotów:**

Wspólne

**Kod przedmiotu:**

1120-IN000-ISP-0041

**Semestr nominalny:**

6 / rok ak. 2015/2016

**Liczba punktów ECTS:**

5

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

1. godziny kontaktowe – 50 h; w tym
a) obecność na wykładach – 30 h
b) obecność na ćwiczeniach – 15 h
c) konsultacje – 5 h
2. praca własna studenta – 75 h; w tym
a) przygotowanie do ćwiczeń i kolokwium końcowego – 50 h
b) zadania domowe – 25 h
Razem 125 h, co odpowiada 5 pkt. ECTS

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

1. obecność na wykładach – 30 h
2. obecność na ćwiczeniach – 15 h
3. konsultacje – 5 h
Razem 50 h, co odpowiada 2 pkt. ECTS

**Język prowadzenia zajęć:**

polski

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

1. obecność na ćwiczeniach – 15 h
2. zadania domowe – 25 h
Razem 40 h, co odpowiada 2 pkt. ECTS

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

|  |  |
| --- | --- |
| Wykład: | 30h |
| Ćwiczenia: | 15h |
| Laboratorium: | 0h |
| Projekt: | 0h |
| Lekcje komputerowe: | 0h |

**Wymagania wstępne:**

Biegła znajomość więcej niż jednego języka programowania wysokiego poziomu, znajomość wyrażeń regularnych i gramatyk bezkon-tekstowych, znajomość podstawowych struktur danych. Przydatna znajomość języka zorientowanego maszynowo (asemblera)
Przedmioty poprzedzające:
Teoria automatów i języków formalnych
Programowanie 2 – obiektowe, Programowanie 3 - zaawansowane
Algorytmy i struktury danych 1
Programowanie w językach zorientowanych maszynowo (zalecane)

**Limit liczby studentów:**

Ćwiczenia – 30 os/grupa

**Cel przedmiotu:**

Celem przedmiotu jest zdobycie wiedzy na temat przebiegu procesu kompilacji, zrozumienie wpływu cech języka źródłowego (wysokiego poziomu) na wydajność generowanego kodu maszynowego oraz nabycie umiejętności przetwarzania tekstów o sformalizowanej strukturze metodami stosowanymi w kompilatorach z wykorzysta-niem popularnych narzędzi. Po ukończeniu kursu studenci powinni:
- znać podstawowe fazy procesu kompilacji (analiza leksykalna, składniowa i semantyczna, generowanie kodu pośredniego i docelowego, optymalizacja) i rozumieć ich znaczenie
- znać podstawy teoretyczne procesu analizy kodu źródłowego
- rozumieć wpływ różnorodnych konstrukcji języków wysokiego poziomu na wydajność generowanego na ich podstawie kodu wynikowego
- rozumieć znaczenie kodu pośredniego i znać podstawowe metody jego optymalizacji
- umieć stosować wybrane narzędzia wspomagające analizę leksykalną i składniową

**Treści kształcenia:**

Program wykładu:
Ogólne pojęcia i fazy procesu kompilacji. Przegląd własności języków programowania. Modele środowiska wykonawczego. Analiza leksykalna. Analiza składniowa i gramatyki bezkontekstowe, w tym metoda zejść rekurencyjnych, analizatory LL(1), analizatory LR(1). Analiza semantyczna i gramatyki atrybutywne. Diagnostyka i obsługa błędów. Generowanie kodu pośredniego i wynikowego. Optymalizacja.
Program ćwiczeń:
Ilustracja materiału z wykładu, przykłady użycia generatorów FLEX i Bison oraz innych narzędzi wspomagających tworzenie kompilatorów.

**Metody oceny:**

Zaliczenie odbywa się łącznie dla wykładu i ćwiczeń. Podstawą zaliczenia jest pisemne kolokwium końcowe. Dodatkowo na ocenę pozytywny wpływ mają nieobowiązkowe zadania domowe i aktywność na ćwiczeniach, a negatywny nieobecności (obecność bez aktywności jest neutralna). Studenci wykazujący się zdecydowanie ponadprzeciętną wiedzą i zaangażowaniem (aktywnością na ćwiczeniach) mogą być zwolnieni z kolokwium końcowego (każdorazowa indywidualna decyzja wykładowcy)

**Egzamin:**

nie

**Literatura:**

1. A. V. Aho, R. Sethi, J. D. Ullman, Kompilatory: Reguły, metody, na-rzędzia, WNT 2002
2. K.C. Louden, Compiler Construction: Principles and Practice, PWS 1997
3. Dokumentacja programów FLEX i Bison

**Witryna www przedmiotu:**

e.mini.pw.edu.pl

**Uwagi:**

## Efekty przedmiotowe

### Profil ogólnoakademicki - wiedza

**Efekt W01:**

Ma uporządkowaną, podbudowaną teoretycznie wiedzę ogólną w zakresie języków i paradygmatów programowania

Weryfikacja:

kolokwium końcowe

**Powiązane efekty kierunkowe:** K\_W06

**Powiązane efekty obszarowe:** T1A\_W03

**Efekt W02:**

Zna podstawowe metody, techniki i narzędzia stosowane przy implementacji języków programowania

Weryfikacja:

kolokwium końcowe

**Powiązane efekty kierunkowe:** K\_W10

**Powiązane efekty obszarowe:** T1A\_W07

### Profil ogólnoakademicki - umiejętności

**Efekt U01:**

Ma umiejętność formułowania algorytmów i ich programowania z użyciem popularnych narzędzi

Weryfikacja:

kolokwium końcowe, zadania domowe

**Powiązane efekty kierunkowe:** K\_U11

**Powiązane efekty obszarowe:** T1A\_U09, T1A\_U14, T1A\_U15