**Nazwa przedmiotu:**

Techniki kompilacji

**Koordynator przedmiotu:**

Rajmund Kożuszek

**Status przedmiotu:**

Fakultatywny ograniczonego wyboru

**Poziom kształcenia:**

Studia I stopnia

**Program:**

Informatyka

**Grupa przedmiotów:**

Inżynieria oprogramowania

**Kod przedmiotu:**

TKOM

**Semestr nominalny:**

6 / rok ak. 2021/2022

**Liczba punktów ECTS:**

4

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

1. liczba godzin kontaktowych – 40 godz., w tym
 obecność na wykładach: 30 godz.,
 obecność na egzaminie: 3 godz.,
 udział w konsultacjach związanych z realizacją wykładu: 2 godz.,
 zajęcia projektowe: konsultowanie i weryfikacja etapów realizacji projektu: 5 godz.
2. praca własna studenta – 70 godz., w tym
 analiza literatury i materiałów wykładowych związana z przygotowaniem do kolejnych wykładów: 30 godz.,
 realizacja projektu: 30 godz.,
 przygotowanie do kolokwiów i egzaminu: 10 godz.

Łączny nakład pracy studenta wynosi 110 godz., co odpowiada 4pkt. ECTS.

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

1,5 pkt. ECTS, co odpowiada 40 godz. kontaktowym

**Język prowadzenia zajęć:**

polski

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

1 pkt. ECTS, co odpowiada 30 godz. realizacji projektu

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

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

**Wymagania wstępne:**

Programowanie obiektowe
Algorytmy i struktury danych
Architektura komputerów
Systemy operacyjne
Sztuka wytwarzania oprogramowania

**Limit liczby studentów:**

60

**Cel przedmiotu:**

Celem przedmiotu jest zapoznanie studentów z głównymi pojęciami, metodami i algorytmami związanymi z transformacją tekstów oraz podstawami budowy kompilatorów i stosowanych w nich metodach i algorytmach. W szczególności, studenci zapoznają się/pogłębiają wiedzę i umiejętności praktyczne w zakresie wyrażeń regularnych (aspektów użytkowych jak i samego sposobu realizacji mechanizmu wyrażeń regularnych – automaty niedeterministyczne i deterministyczne), gramatyki i języki (metody specyfikacji, różne klasy gramatyk – metody weryfikacji przynależności do danej klasy, przekształcanie gramatyk), gramatyki bezkontekstowe, teoretyczne i praktyczne aspekty analizy leksykalnej, składniowej, semantycznej, interpretacja i generacja kodu. W sposób praktyczny student zdobywa i weryfikuje swoją wiedzę i umiejętności poprzez realizację, w ramach zajęć projektowych, np. interpretera własnego języka programowania czy fragmentu kompilatora.

**Treści kształcenia:**

1. Wykład:
Wprowadzenie (1h):
Ogólnie o problemie kompilacji / interpretacji; wprowadzenie do budowy kompilatorów i procesu kompilacji; składniki zależne od języka źródłowego (front end) i od platformy docelowej. Potrzebne techniki i narzędzia (również teoretyczne);

Języki regularne i analiza leksykalna (5h):
Reprezentacje języków regularnych, formalizm wyrażeń regularnych (WR), gramatyki regularne; automaty niedeterministyczne (AN) i deterministyczne (AD). Konwersja wyrażeń regularnych na automaty, algorytm Thompsona, konwersja AN na AD (algorytm podzbiorowy), bezpośrednia konwersja WR na AD. Właściwości klasy języków regularnych. Zastosowania do wyszukiwania wzorców i analizy leksykalnej. Przykład implementacji analizatora leksykalnego i obsługi źródeł. Generatory analizatorów leksykalnych.

Makrogeneracja – parametryczne przetwarzanie tekstów (3h):
Typy substytucji tekstowych, rozpoznawanie makrodefinicji i makrowywołań, organizacja biblioteki makrodefinicji, reguły przesłaniania i dostępności; mechanizmy wiązania parametrów. Przykłady makrogeneratorów.

Podstawy teoretyczne – języki i gramatyki (3h):
Języki i gramatyki; style definiowania języka; konwencje notacyjne; przegląd metanotacji (BNF, EBNF, ISO-14977, ABNF, diagramy składniowe); Gramatyki generacyjne; Hierarchia języków wg Chomskiego, formy kanoniczne gramatyk (CNF, GNF, KNF); Twierdzenie o pompowaniu dla gramatyk bezkontekstowych; Drzewa wyprowadzeń dla gramatyk bezkontekstowych. Niejednoznaczność w gramatykach bezkontekstowych - przykłady.

Języki bezkontekstowe i ich rozbiór (2h):
Reprezentacje języków bezkontekstowych; właściwości gramatyk bezkontekstowych. Przekształcanie gramatyk (substytucja, usuwanie symboli niedostępnych, usuwanie produkcji pustych i jednostkowych); Usuwanie rekursji lewostronnej, faktoryzacja lewostronna. Zbiory FIRST i FOLLOW;

Rozbiór rekursywnie zstępujący (5h).
Schematy translacji. Schemat translacji wyrażeń arytmetycznych. Rozbiór rekursywnie zstępujący sterowany składnią (RD); Rekursywnie zstępujący analizator składniowy – przykłady praktyczne; struktury pomocnicze w analizie składniowej; Współpraca analizator składniowy – analizator leksykalny; Obsługa błędów; Rozbiór sterowany tablicą - parser LL(1);

Analizatory składniowe wstępujące (5h):
Schemat ogólny metod wstępujących; LR-formy; Automat LR(0) i sterownik parsera LR(0), rozbiór wyrażeń wg LR(0). Konflikty w tabelach LR(0), rozbiór SLR(1), konflikty w SLR(1). Rozbiór LR(1), LR1-formy, wyznaczanie tabeli dla parsera LR(1), redukcja LR(1) - LALR(1). Porównanie parserów. Własności i problemy decyzyjne języków bezkontekstowych.

Gramatyki z atrybutami i operatorowe (2h)
Gramatyki operatorowe; Rozbiór wyrażeń w gramatykach operatorowych; przykład translacji wyrażeń do Odwrotnej Notacji Polskiej (ONP); Gramatyki z atrybutami i ich wykorzystanie w generatorach parserów; Akcje semantyczne parsera.

Analiza semantyczna (2h):
Struktury danych i schematy programistyczne w kompilatorze; Atrybuty identyfikatorów, organizacja tabel symboli; organizacja rekursji i zmiennych lokalnych na przykładzie interpreterów.

Generacja kodu i techniki optymalizacyjne (2h):
Komunikacja analizatora z generatorem, alokacja pamięci w strukturach blokowych, generacja kodu dla wyrażeń i struktur sterowania; przejście z konwencji maszyny stosowej do konwencji procesora docelowego. Elementy optymalizacji kodu, optymalizacje lokalne i globalne; algorytm optymalizacji wyrażeń.

2. Projekt
Zakres projektu
Celem projektu jest zapoznanie się z metodami wytwarzania i budową kompilatorów. W szczególności, opanowanie przez studenta praktycznych umiejętności realizacji przetwarzania sterowanego składnią w odniesieniu do różnych typów zastosowań wykorzystujących notację sformalizowaną (symulacja, przetwarzanie i rozpoznawanie tekstu, interpretacja prostych języków). Wymagane jest przygotowanie projektu (w szczególności: zebranie i wyspecyfikowanie wymagań w toku dyskusji z prowadzącym projekt, opracowanie gramatyki własnego języka, koncepcji rozwiązania) oraz dokumentacji końcowej.

**Metody oceny:**

Realizacja przedmiotu obejmuje następujące formy zajęć:
• wykład prowadzony w wymiarze 2 godz. tygodniowo (30 godzin); w wybranych zagadnieniach przewidziana jest aktywizacja studentów na wykładzie;
• zajęcia projektowe (30 godzin); w ramach tych zajęć student, w oparciu o ustalane z prowadzącym wymagania funkcjonalne, opracowuje składnię i semantykę własnego języka a następnie realizuje kompilator, interpreter lub inny procesor tekstu. Przygotowuje projekt wstępny (zawierający zebrane wymagania, projekt rozwiązania i przypadki testowe), który po akceptacji prowadzącego jest implementowany w uzgodnionym z prowadzącym języku programowania i testowany. Efekty pracy student demonstruje w laboratorium. Po prezentacji programu student opracowuje dokumentacje końcową i przedstawia ją prowadzącemu. Student może ponadto uczestniczyć w prowadzonych co tydzień w wymiarze 2 godz. konsultacjach.

Sprawdzanie założonych efektów kształcenia realizowane jest przez:
• ocenę wiedzy i umiejętności związanych z realizacją tematów projektowych;
• formatywną ocenę związaną z rozwiązaniem zadań przedkolokwialnych, a także z interaktywną forma prowadzenia wykładu oraz z ćwiczeniami;
• ocenę wiedzy i umiejętności wykazanych na kolokwiach (na kolokwiach student może korzystać z wybranych materiałów dydaktycznych oraz książek);
• ocenę wiedzy i umiejętności wykazanych na egzaminie (student może korzystać z wybranych materiałów dydaktycznych oraz książek).

**Egzamin:**

tak

**Literatura:**

• Aho A.V., Lam M., Sethi R., Ullman J.D.: “Compilers: Principles, Techniques & Tools”, 2nd edition, Pearson, Addison-Wesley, 2013;
• Aho A.V., Sethi R., Ullman J.D.: Kompilatory: reguły, metody i narzędzia, WNT 2006;
• Bennet J.P.: “Introduction to Compiling Techniques: A First Course Using ANSI C, Lex, and Yacc”, 2nd edition, Mc Graw Hill, 1996;
• Grune D., van Reeuwijk K., Bal H. E., Jacobs C. J. H., Langendoen K.: “Modern Compiler Design”, Springer, 2016
• Reinhard W.: “Compiler Design”, Springer-Verlag Berlin and Heidelberg, 2013;
• K.D. Cooper, L.Torcon: “Engineering a Compiler”, 2nd edition, Elsevier, 2011;
• Mak R.:“Writing Compilers and Interpreters: A Modern Software Engineering Approach Using Java”, 2014;
• Galles D.: “Modern Compiler Design”, Addison-Wesley, 2004;
• Aho A.V., Sethi R., Ullman J.D., Lam J.D.: „21st Century Compilers”, Addison-Wesley, 2006;
• M. T. Aegidius: “Introduction to Compiler Design”, Springer, 2011;
• Pająk A., Wigura A.: Makrogeneratory, asemblery i konsolidatory, PWN, 1983;
• Strona domowa i dokumentacja projektu ANTLR - https://www.antlr.org ;
• Materiały elektroniczne na stronie przedmiotu

**Witryna www przedmiotu:**

https://usosweb.usos.pw.edu.pl/kontroler.php?\_action=katalog2/przedmioty/pokazPrzedmiot&prz\_kod=103D-INIOP-ISP-TKOM

**Uwagi:**

(-)

## Charakterystyki przedmiotowe

### Profil ogólnoakademicki - wiedza

**Charakterystyka W01:**

zna metody opisu języków formalnych

Weryfikacja:

kolokwia, egzamin, projekt

**Powiązane charakterystyki kierunkowe:** W05

**Powiązane charakterystyki obszarowe:** P6U\_W, I.P6S\_WG.o

**Charakterystyka W02:**

zna hierarchię języków wg Chomskiego i ich modeli generacyjnych / obliczeniowych

Weryfikacja:

kolokwia, egzamin, projekt

**Powiązane charakterystyki kierunkowe:** W05

**Powiązane charakterystyki obszarowe:** P6U\_W, I.P6S\_WG.o

**Charakterystyka W03:**

zna mechanizmy makrogeneracji i ich dostępność w środowiskach programistycznych

Weryfikacja:

kolokwia, egzamin, projekt

**Powiązane charakterystyki kierunkowe:** W08

**Powiązane charakterystyki obszarowe:** P6U\_W, I.P6S\_WG.o, III.P6S\_WG

**Charakterystyka W04:**

zna metody reprezentacji i przetwarzania języków regularnych

Weryfikacja:

kolokwia, egzamin, projekt

**Powiązane charakterystyki kierunkowe:** W05

**Powiązane charakterystyki obszarowe:** P6U\_W, I.P6S\_WG.o

**Charakterystyka W05:**

zna metody analizy jednoznacznych języków bezkontekstowych (parser rekursywnie zstępujący, rozbiór wstępujący)

Weryfikacja:

kolokwia, egzamin, projekt

**Powiązane charakterystyki kierunkowe:** W05

**Powiązane charakterystyki obszarowe:** P6U\_W, I.P6S\_WG.o

**Charakterystyka W06:**

zna podstawowe metody uwzględniania semantyki języka i generowania kodu wynikowego

Weryfikacja:

kolokwia, egzamin, projekt

**Powiązane charakterystyki kierunkowe:** W08, W09

**Powiązane charakterystyki obszarowe:** III.P6S\_WG, P6U\_W, I.P6S\_WG.o

**Charakterystyka W07:**

zna metody rozstrzygalności głównych problemów decyzyjnych w klasach języków regularnych i bezkontekstowych

Weryfikacja:

kolokwia, egzamin, projekt

**Powiązane charakterystyki kierunkowe:** W05

**Powiązane charakterystyki obszarowe:** P6U\_W, I.P6S\_WG.o

**Charakterystyka W08:**

zna zasady działania kompilatorów i interpreterów

Weryfikacja:

kolokwia, egzamin, projekt

**Powiązane charakterystyki kierunkowe:** W08, W09

**Powiązane charakterystyki obszarowe:** P6U\_W, I.P6S\_WG.o, III.P6S\_WG

**Charakterystyka W09:**

zna metody i rozumie potrzebę modularyzacji przetwarzania danych

Weryfikacja:

kolokwia, egzamin, projekt

**Powiązane charakterystyki kierunkowe:** W08, W09

**Powiązane charakterystyki obszarowe:** P6U\_W, I.P6S\_WG.o, III.P6S\_WG

**Charakterystyka W10:**

rozumie niebezpieczeństwa i wady nieodpowiedniego przetwarzania danych wejściowych w programach

Weryfikacja:

kolokwia, egzamin, projekt

**Powiązane charakterystyki kierunkowe:** W08, W10

**Powiązane charakterystyki obszarowe:** P6U\_W, I.P6S\_WG.o, III.P6S\_WG

**Charakterystyka W11:**

rozumie wady i zalety stosowania wyrażeń regularnych oraz ich implementację w dostępnych bibliotekach i językach

Weryfikacja:

kolokwia, egzamin, projekt

**Powiązane charakterystyki kierunkowe:** W08, W05

**Powiązane charakterystyki obszarowe:** III.P6S\_WG, P6U\_W, I.P6S\_WG.o

### Profil ogólnoakademicki - umiejętności

**Charakterystyka U01:**

potrafi zdefiniować formalnie składnię języka w metanotacji EBNF (lub podobnej)

Weryfikacja:

kolokwia, egzamin, projekt

**Powiązane charakterystyki kierunkowe:** U01, U04, U09

**Powiązane charakterystyki obszarowe:** P6U\_U, I.P6S\_UW.o, III.P6S\_UW.o, I.P6S\_UK

**Charakterystyka U02:**

potrafi przekształcić gramatykę języka wg zadanych kryteriów (sprawdzić kryteria LL(1), usunąć produkcje jednostkowe, sprowadzić do postaci normalnej Chomskiego)

Weryfikacja:

kolokwia, egzamin, projekt

**Powiązane charakterystyki kierunkowe:** U04, U07

**Powiązane charakterystyki obszarowe:** P6U\_U, I.P6S\_UW.o, III.P6S\_UW.o

**Charakterystyka U03:**

potrafi przekształcać dowolnie reprezentację języka regularnego (wyrażenia regularne, automaty skończone, gramatyki regularne)

Weryfikacja:

kolokwia, egzamin, projekt

**Powiązane charakterystyki kierunkowe:** U07

**Powiązane charakterystyki obszarowe:** P6U\_U, I.P6S\_UW.o, III.P6S\_UW.o

**Charakterystyka U04:**

potrafi zrealizować analizator leksykalny i parser rekursywnie zstępujący wg zadanej gramatyki

Weryfikacja:

kolokwia, egzamin, projekt

**Powiązane charakterystyki kierunkowe:** U04

**Powiązane charakterystyki obszarowe:** P6U\_U, I.P6S\_UW.o, III.P6S\_UW.o

**Charakterystyka U05:**

potrafi zastosować model przetwarzania sterowanego składnią (schemat translacji) do prostych problemów obliczeniowych

Weryfikacja:

kolokwia, egzamin, projekt

**Powiązane charakterystyki kierunkowe:** U10, U04, U08

**Powiązane charakterystyki obszarowe:** III.P6S\_UW.o, P6U\_U, I.P6S\_UW.o, I.P6S\_UO, I.P6S\_UK

**Charakterystyka U06:**

potrafi zaprojektować środowisko programowe do osadzenia akcji semantycznych dla prostego języka bezkontekstowego

Weryfikacja:

kolokwia, egzamin, projekt

**Powiązane charakterystyki kierunkowe:** U01

**Powiązane charakterystyki obszarowe:** P6U\_U, I.P6S\_UW.o, III.P6S\_UW.o

**Charakterystyka U07:**

potrafi wykorzystywać generatory analizatorów leksykalnych i składniowych oraz integrować je we własnym programie

Weryfikacja:

kolokwia, egzamin, projekt

**Powiązane charakterystyki kierunkowe:** U04, U10

**Powiązane charakterystyki obszarowe:** P6U\_U, I.P6S\_UW.o, III.P6S\_UW.o, I.P6S\_UK

**Charakterystyka U08:**

potrafi odpowiednio ustrukturalizować kod programu i schemat przetwarzania

Weryfikacja:

kolokwia, egzamin, projekt

**Powiązane charakterystyki kierunkowe:** U02, U06

**Powiązane charakterystyki obszarowe:** P6U\_U, I.P6S\_UW.o, III.P6S\_UW.o

**Charakterystyka U09:**

potrafi stosować dobre wzorce programowania do przetwarzania tekstu

Weryfikacja:

kolokwia, egzamin, projekt

**Powiązane charakterystyki kierunkowe:** U02, U10

**Powiązane charakterystyki obszarowe:** P6U\_U, I.P6S\_UW.o, III.P6S\_UW.o, I.P6S\_UK

**Charakterystyka U10:**

potrafi w sposób bezpieczny implementować strumieniowe przetwarzanie żądań, tekstu itp.

Weryfikacja:

kolokwia, egzamin, projekt

**Powiązane charakterystyki kierunkowe:** U10, U02, U06

**Powiązane charakterystyki obszarowe:** P6U\_U, I.P6S\_UK, III.P6S\_UW.o, I.P6S\_UW.o

**Charakterystyka U11:**

potrafi sprawnie posługiwać się wyrażeniami regularnymi

Weryfikacja:

kolokwia, egzamin, projekt

**Powiązane charakterystyki kierunkowe:** U02, U06

**Powiązane charakterystyki obszarowe:** P6U\_U, I.P6S\_UW.o, III.P6S\_UW.o

### Profil ogólnoakademicki - kompetencje społeczne

**Charakterystyka K01:**

rozumie potrzebę stałego aktualizowania i wzbogacania posiadanej wiedzy

Weryfikacja:

kolokwia, egzamin, projekt

**Powiązane charakterystyki kierunkowe:** K01, K03

**Powiązane charakterystyki obszarowe:** P6U\_K, I.P6S\_KK, I.P6S\_KR

**Charakterystyka K02:**

ma świadomość konieczności komunikowania się z otoczeniem, także pozazawodowym, w sposób zrozumiały dla odbiorcy

Weryfikacja:

kolokwia, egzamin, projekt

**Powiązane charakterystyki kierunkowe:** K05

**Powiązane charakterystyki obszarowe:** P6U\_K, I.P6S\_KO

**Charakterystyka K03:**

potrafi planować działania projektowe wg wymaganego terminu

Weryfikacja:

kolokwia, egzamin, projekt

**Powiązane charakterystyki kierunkowe:** K03

**Powiązane charakterystyki obszarowe:** P6U\_K, I.P6S\_KK, I.P6S\_KR

**Charakterystyka K04:**

potrafi samodzielnie pozyskiwać poszerzające informacje o rozwiązywanym problemie i dostępnych narzędziach programowych

Weryfikacja:

kolokwia, egzamin, projekt

**Powiązane charakterystyki kierunkowe:** K01, K03

**Powiązane charakterystyki obszarowe:** P6U\_K, I.P6S\_KK, I.P6S\_KR