

<b>KARTA OPISU MODUŁU KSZTAŁCENIA</b>			
Nazwa modułu/przedmiotu <b>Formal Languages and Compilers</b>			Kod <b>1010511341010510193</b>
Kierunek studiów <b>Informatyka</b>		Profil kształcenia (ogólnoakademicki, praktyczny) <b>ogólnoakademicki</b>	Rok / Semestr <b>2 / 4</b>
Ścieżka obieralności/specjalność <b>-</b>		Przedmiot oferowany w języku: <b>angielski</b>	Kurs (obligatoryjny/obieralny) <b>obieralny</b>
Stoień studiów: <b>I stopień</b>		Forma studiów (stacjonarna/niestacjonarna) <b>stacjonarna</b>	
Godziny Wykłady: <b>30</b> Ćwiczenia: <b>-</b> Laboratoria: <b>15</b> Projekty/seminaria: <b>-</b>			Liczba punktów <b>5</b>
Status przedmiotu w programie studiów (podstawowy, kierunkowy, inny) <b>kierunkowy</b>		(ogólnouczelniany, z innego kierunku) <b>z danego kierunku</b>	
Obszar(y) kształcenia i dziedzina(y) nauki i sztuki <b>nauki techniczne</b>  <b>nauki techniczne</b>			Podział ECTS (liczba i %) <b>5 100%</b>  <b>5 100%</b>
<b>Odpowiedzialny za przedmiot / wykładowca:</b>			
dr inż. W. Complik email: Wojciech.Complik@cs.put.poznan.pl tel. (0-61) 665-2983 Wydział Informatyki ul. Piotrowo 3, 60-965 Poznań		dr hab. inż. Jerzy Nawrocki email: Jerzy.Nawrocki@cs.put.poznan.pl tel. (0-61) 665-3422 Wydział Informatyki ul. Piotrowo 3, 60-965 Poznań	
<b>Wymagania wstępne w zakresie wiedzy, umiejętności, kompetencji społecznych:</b>			
1	<b>Wiedza:</b>	Student rozpoczynający ten przedmiot powinien posiadać podstawową wiedzę z algorytmiki i programowania w językach imperatywnych.	
2	<b>Umiejętności:</b>	Powinien posiadać umiejętność rozwiązywania podstawowych problemów z zakresu projektowania, sprawdzenia poprawności i implementowania algorytmów w języku C oraz umiejętność pozyskiwania informacji ze wskazanych źródeł.	
3	<b>Kompetencje społeczne</b>	Powinien również rozumieć konieczność poszerzania swoich kompetencji / mieć gotowość do podjęcia współpracy w ramach zespołu. Ponadto w zakresie kompetencji społecznych student musi prezentować takie postawy jak uczciwość, odpowiedzialność, wytrwałość, ciekawość poznawcza, kreatywność, kultura osobista, szacunek dla innych ludzi.	
<b>Cel przedmiotu:</b>			
1. Przekazanie studentom podstawowej wiedzy z zakresu praktycznych aspektów teorii języków formalnych oraz budowy translatorów i środowisk czasu wykonania w zakresie zasad, technik i narzędzi wykorzystywanych wspólnie do budowy kompilatorów i innych narzędzi do automatycznego przetwarzania tekstu, takich jak: edytory tekstu, systemy wyszukiwania informacji, systemy składu elektronicznego i weryfikatory programów.			
2. Rozwijanie u studentów umiejętności rozwiązywania prostych problemów za pomocą języków programowania ogólnego przeznaczenia jak i z wykorzystaniem do tego celu specjalistycznych narzędzi. Poszerzenie wiedzy na temat wcześniej wykorzystywanych środowisk programistycznych i języków programowania w wyniku spojrzenia na nie z punktu widzenia projektanta i implementatora a nie tylko użytkownika.			
<b>Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia</b>			
<b>Wiedza:</b>			
1. ma rozszerzoną i pogłębioną wiedzę z matematyki przydatną do formułowania i rozwiązywania złożonych zadań informatycznych dotyczących formalnej specyfikacji i weryfikacji oprogramowania - [K1st_W1]			
2. ma uporządkowaną i podbudowaną teoretycznie wiedzę ogólną w zakresie języków i paradygmatów programowania - [K1st_W4]			
3. zna podstawowe techniki, metody i narzędzia wykorzystywane w procesie rozwiązywania zadań informatycznych w zakresie analizy algorytmów i implementacji języków programowania - [K1st_W7]			
<b>Umiejętności:</b>			
1. potrafi właściwie zaplanować i wykonać testy funkcjonalne i pozafunkcjonalne oprogramowania - [K1st_U3]			
2. potrafi zaprojektować oraz zrealizować zadanie informatyczne stosując odpowiednio dobrane metody analityczne i eksperymentalne - [K1st_U4]			
3. ma umiejętność specyfikowania i implementowania analizatorów z wykorzystaniem poznanych narzędzi - [K1st_U11]			
<b>Kompetencje społeczne:</b>			

1. ma świadomość znaczenia wiedzy z zakresu języków formalnych i kompilatorów w rozwiązywaniu problemów inżynierskich - [K1st\_K2]

### Sposoby sprawdzenia efektów kształcenia

Efekty kształcenia przedstawione wyżej weryfikowane są w następujący sposób:

Ocena formująca:

a) w zakresie wykładów:

- na podstawie odpowiedzi na pytania dotyczące materiału omówionego na poprzednich wykładach;

b) w zakresie ćwiczeń:

- na podstawie oceny bieżącego postępu realizacji zadań,

Ocena podsumowująca:

Sprawdzanie założonych efektów kształcenia realizowane jest przez:

- ocenę przygotowania studenta do poszczególnych sesji zajęć laboratoryjnych (sprawdzian 'wejściowy') oraz ocenę umiejętności związanych z realizacją ćwiczeń laboratoryjnych,

- ocenianie ciągle, na każdych zajęciach (odpowiedzi ustne) - premiowanie przyrostu umiejętności posługiwania się poznanymi zasadami i metodami,

- ocenę wiedzy i umiejętności związanych z realizacją zadań projektowych/laboratoryjnych poprzez kolokwium pod koniec semestru, kolokwium obejmuje 11 zadań o charakterze praktycznym dotyczących poszczególnych narzędzi i zagadnień omawianych w ramach przedmiotu (2 x AWK, 2 x lex, 2 x LLgen, 2 x yacc, 3 x SLR); zadania mają zarówno charakter konstrukcyjny (np. napisz program) jak i analityczny (np. jaka będzie odpowiedź danego programu)

- ocenę i 'obronę' przez studenta sprawozdania z realizacji projektu,

Uzyskiwanie punktów dodatkowych za aktywność podczas zajęć, a szczególnie za:

- omówienia dodatkowych aspektów zagadnienia,

- efektywność zastosowania zdobytej wiedzy podczas rozwiązywania zadanego problemu,

- uwagi związane z udoskonaleniem materiałów dydaktycznych,

- wskazywanie trudności percepcyjnych studentów umożliwiające bieżące doskonalenie procesu dydaktycznego.

### Treści programowe

Pierwszy wykład poświęcony jest omówieniu organizacji zajęć (zakresu przedmiotu, środowiska i narzędzi, literatury i zasad zaliczania) oraz wprowadzeniu do tematyki przetwarzania tekstu na przykładzie języka AWK.

Na drugim wykładzie przedstawiany jest model analiza-synteza translatora, podział procesu translacji na etapy oraz faza analizy leksykalnej i zasady prowadzenie jej z wykorzystaniem generatora analizatorów leksykalnych lex.

Pierwsze zajęcia laboratoryjne poświęcone są zagadnieniom organizacyjnym: zaznajomieniu się ze środowiskiem i narzędziami, uruchamianiem skryptów do kompilacji oraz nauce wykorzystywania języka AWK do przetwarzania tekstu.

Trzeci wykład, otwierający cykl poświęcony analizie składniowej, zawiera omówienie ogólnych zasad prowadzenia analizy składniowej i pojęć związanych z gramatykami bezkontekstowymi (takich jak: terminale i nieterminale, produkcje, wywody, typy rekurencji, niejednoznaczność i równoważność gramatyki) oraz wstęp do metody zstępującej.

W dalszym ciągu cyklu poświęconego analizie składniowej prezentowany jest generator analizatorów składniowych działających w oparciu o metodę zstępującą - LLgen. W pierwszym wykładzie poświęconym temu generatorowi prezentowane są ogólne zasady jego działania i konstruowania specyfikacji analizatorów syntaktycznych.

W ramach zajęć laboratoryjnych studenci przechodzą do zapoznawania się z projektowaniem i implementowaniem prostych filtrów tekstu z wykorzystaniem generatora analizatorów leksykalnych lex.

W trakcie kolejnego wykładu przedstawiana jest koncepcja translacji sterowanej składnią. Przedstawiane są pojęcia atrybutów, definicji sterowanych składnią, schematów translacji oraz definicji S-atrybutowych i L-atrybutowych. Omawiane są również zasady implementacji translacji sterowanej składnią w generatorze LLgen.

Szósty wykład poświęcony jest metodzie wstępującej, zasadom konstruowania i działania analizatorów działających tą metodą i generatorowi yacc. Wykład obejmuje charakterystykę generatora yacc, składnię specyfikacji analizatora składniowego, zasady współpracy z analizatorem leksykalnym oraz wykrywania i obsługi błędów składniowych.

W trakcie zajęć laboratoryjnych studenci, implementując proste filtry tekstu, zapoznają się z podstawami generatora LLgen i zasadami łączenia analizatorów składniowych i leksykalnych.

W ramach siódmego wykładu przedstawiane są zasady implementacji translacji sterowanej składnią w metodzie wstępującej w generatorze yacc (atrybuty syntetyzowane i dziedziczone, typy atrybutów, akcje wielokrotne).

Kolejny wykład z cyklu dotyczącego analizy składniowej poświęcony jest posługiwaniu się gramatykami niejednoznaczными w metodzie wstępującej w generatorze yacc. Przedstawiane są zalety i typowe, praktyczne przykłady gramatyk niejednoznacznych oraz zasady wykorzystywania ich w generatorze yacc.

W czasie zajęć laboratoryjnych studenci samodzielnie implementują z wykorzystaniem generatorów LLgen i lex analizator prostego języka programowania imperatywnego.

W ramach dziewiątego wykładu przedstawiana jest analiza semantyczna: różne typy kontroli zależności kontekstowych, takie jak: sprawdzenie przepływu sterowania, unikalności deklaracji nazw, powtórzeń nazw oraz kontrola typów.

Wykład kończący cykl dotyczący analizy składniowej poświęcony jest porównaniu wad i zalet różnych metod tworzenia translatorów działających w oparciu o metodę wstępującą oraz demonstracji sposobu generowania kodu parsersa.

Na zajęciach laboratoryjnych studenci rozwiązują zadania dotyczące generatora yacc.

Kolejny wykład poświęcony jest etapowi syntezy kodu pośredniego: omawiane są różne rodzaje kodów pośrednich i maszyny wirtualne, a jako przykład konkretnej implementacji, szczegółowo przedstawiany jest kod trójadresowy.

Dwunasty wykład dotyczy generacji kodu wynikowego i jego optymalizacji oraz zagadnień budowy środowiska wykonawczego, takich jak dostęp do nazw nielokalnych, dynamiczny przydział pamięci i przekazywanie parametrów do podprogramów.

Na laboratoriach studenci rozpoczynają implementację w yaccu i lexie translatora tłumaczącego kod pomiędzy wybranymi, znanymi im językami imperatywnymi.

Ostatnim prezentowanym na wykładach narzędziem jest zintegrowane środowisko generatora ANTLR.

Na zajęciach laboratoryjnych studenci kończą implementację i przeprowadzają testy translatorów.

Ostatni wykład poświęcony jest podsumowaniu całości przedstawionych w ciągu semestru zagadnień oraz sprawdzeniu wiedzy studentów w formie kolokwium zaliczeniowego.

Na zajęciach laboratoryjnych studenci rozliczają wykonanie projektów.

Metody dydaktyczne:

1. wykład: prezentacja multimedialna, prezentacja ilustrowana przykładami podawanymi na tablicy, rozwiązywanie zadań, demonstracja narzędzi programistycznych,

2. ćwiczenia laboratoryjne: rozwiązywanie zadań, dyskusja.

#### Literatura podstawowa:

1. Kompilatory. Reguły, metody i narzędzia, A. V. Aho, R. Sethi, J. D. Ullman, WNT, Warszawa, 2002
2. Automatyczne przetwarzanie tekstów, J. Cybulka, B. Jankowska, J. Nawrocki, Nakom, Poznań, 2002
3. Wprowadzenie do przetwarzania tekstów w języku AWK, J. Nawrocki, W. Complak, Nakom (Pro Dialog), Poznań, 1994
4. Wprowadzenie do generatora Lex, J. Nawrocki, A. Czajka, Nakom (Pro Dialog), Poznań, 1998
5. Kompilatory. Reguły, metody i narzędzia, A. V. Aho, R. Sethi, J. D. Ullman, WNT, Warszawa, 2002
6. Automatyczne przetwarzanie tekstów, J. Cybulka, B. Jankowska, J. Nawrocki, Nakom, Poznań, 2002
7. Wprowadzenie do przetwarzania tekstów w języku AWK, J. Nawrocki, W. Complak, Nakom (Pro Dialog), Poznań, 1994
8. Wprowadzenie do generatora Lex, J. Nawrocki, A. Czajka, Nakom (Pro Dialog), Poznań, 1998

#### Literatura uzupełniająca:

1. lex & yacc, 2nd Edition, D. Brown, J. Levine, T. Mason, O'Reilly Media, 1992
2. The Definitive ANTLR Reference: Building Domain-Specific Languages, T. Parr, The Pragmatic Bookshelf, 2007
3. Compilers: Principles, Techniques, and Tools, 2. Ed., A.V. Aho, M. S. Lam, R. Sethi, J.D. Ullman, Addison-Wesley, 2007
4. lex & yacc, 2nd Edition, D. Brown, J. Levine, T. Mason, O'Reilly Media, 1992
5. The Definitive ANTLR Reference: Building Domain-Specific Languages, T. Parr, The Pragmatic Bookshelf, 2007
6. Compilers: Principles, Techniques, and Tools, 2. Ed., A.V. Aho, M. S. Lam, R. Sethi, J.D. Ullman, Addison-Wesley, 2007

#### Bilans nakładu pracy przeciętnego studenta

Czynność	Czas (godz.)
1. udział w zajęciach laboratoryjnych / ćwiczeniach	15
2. przygotowanie do ćwiczeń laboratoryjnych	30
3. udział w konsultacjach związanych z realizacją procesu kształcenia, w szczególności ćwiczeń laboratoryjnych / projektu	2
4. napisanie programu / programów, uruchomienie i weryfikacja (czas poza zajęciami laboratoryjnymi)	10
5. przygotowanie do sprawdzianów	10
6. udział w wykładach	30
7. zapoznanie się ze wskazaną literaturą / materiałami dydaktycznymi (10 stron tekstu naukowego = 1 godz.), 100 stron	10
8. przygotowanie do egzaminu i udział w egzaminie	15

#### Obciążenie pracą studenta

forma aktywności	godzin	ECTS
Łączny nakład pracy	123	5
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	49	2
Zajęcia o charakterze praktycznym	55	2