

METODY KONSTRUKCJI KOMPILATORÓW
 Compiler Construction Methods
 Rok 2014/2015 Semestr zimowy

Nazwa modułu kształcenia	Implementacja języków i optymalizacja kodu		
Nazwa jednostki prowadzącej moduł	Instytut Informatyki, Wydział Matematyki i Informatyki		
Kod modułu	WMI.II-IJOK-MI		
Język kształcenia	Polski		
Efekty kształcenia dla modułu kształcenia	Symbol	Efekty kształcenia	Odniesienie do efektów kierunkowych
	E1	zna podstawowe metody konstrukcji kompilatorów i metody ich realizacji programistycznej	K_W14++, K_W10++
	E2	zna wybrane metody opisu składni i semantyki języków programowania	K_W10+, K_W03++
	E3	zna podstawowe narzędzia wspomagające implementacje modułów kompilatorów (Flex, Lex, Yacc, Bison)	K_W05+
	E4	zna wybrane metody optymalizacji kodu	K_W07+
	E5	posługuje się wyrażeniami regularnymi i gramatykami bezkontekstowymi przy opisie atomów leksykalnych i składni języków programowania	K_U15+
	E6	projektuje i implementuje moduły kompilatora wykorzystując podstawowe struktury danych oraz wybrane techniki programistyczne	K_U05++, K_U14+, K_U15+, K_U16+
	E7	potrafi wykorzystywać narzędzia wspomagające generowanie modułów kompilatorów	K_U09+, K_U06+
	E8	prezentuje opracowanie projektu i implementacji kompilatora prostego języka	K_U16+, K_U21+, K_U23+
Typ modułu kształcenia (obowiązkowy/fakultatywny)	fakultatywny		
Rok studiów			
Semestr	zimowy		
Imię i nazwisko osoby/osób prowadzących moduł	dr Jan Rosek		

METODY KONSTRUKCJI KOMPILATORÓW
Compiler Construction Methods
Rok 2014/2015 Semestr zimowy

Imię i nazwisko osoby/osób egzaminującej/egzaminujących bądź udzielającej zaliczenia, w przypadku gdy nie jest to osoba prowadząca dany moduł	dr Jan Rosek
Sposób realizacji	wykład, laboratorium
Wymagania wstępne i dodatkowe	Metody programowania, Algorytmy i struktury danych, Teoria języków i automatów
Liczba godzin zajęć dydaktycznych wymagających bezpośredniego udziału nauczyciela akademickiego i studentów, gdy w danym module przewidziane są takie zajęcia	60
Liczba punktów ECTS przypisana modułowi	6
Bilans punktów ECTS	<ol style="list-style-type: none"> 1. Udział w wykładach - 30 godz. 2. Udział w zajęciach laboratoryjnych – 30 godz. 3. Projektowanie i samodzielna implementacja modułów kompilatorów – 60 godz. 4. Samodzielne rozwiązywanie zadań tablicowych– 30 godz. 5. Przygotowanie do kolokwiów i egzaminu oraz obecność na egzaminie – 30 godz. 6. Łączny nakład pracy studenta: 180 godzin , co odpowiada 6 punktom ECTS
Stosowane metody dydaktyczne	<ol style="list-style-type: none"> 1. Wykład ilustrowany prezentacją komputerową. 2. Ćwiczenia w laboratorium komputerowym, połączone z dyskusją przy tablicy obejmują: <ol style="list-style-type: none"> a. Projektowanie w zespołach przykładowego języka programowania, b. Projektowanie poszczególnych modułów kompilatora c. Prezentacja wyników zakończonych etapów pracy.

METODY KONSTRUKCJI KOMPILATORÓW
Compiler Construction Methods
Rok 2014/2015 Semestr zimowy

	<ol style="list-style-type: none"> 3. Samodzielna implementacja modułów kompilatora i ich testowanie . 4. Prezentacja implementacji zrealizowanych kompilatorów.
<p>Metody sprawdzania i oceny efektów kształcenia uzyskanych przez studentów</p>	<ol style="list-style-type: none"> 1. Dwa kolokwia (E1, E2, E3, E4, E5, E6, E7) 2. Egzamin (E1, E2, E3, E4, E5, E6, E7) 3. Ocena projektu modułów kompilatora (E1, E2, E4, E5, E8) 4. Ocena poprawności działania modułów kompilatora (E5, E6, E7, E8)
<p>Forma i warunki zaliczenia modułu, w tym zasady dopuszczenia do egzaminu, zaliczenia, a także forma i warunki zaliczenia poszczególnych zajęć wchodzących w zakres danego modułu</p>	<p>Student otrzymuje ocenę końcową z ćwiczeń na podstawie: 40 pkt - za projekt i implementację modułów kompilatora, 40 pkt – za dwa kolokwia 20 pkt – za aktywność (obecność i ocena rozwiązań zadań) Warunkiem otrzymania zaliczenia ćwiczeń jest uzyskanie co najmniej połowy punktów</p> <p>Student otrzymuje ocenę końcową z egzaminu na podstawie punktów: 20 pkt – za obecność na wykładzie 80 pkt – za egzamin ustny.</p> <p>Warunkiem dopuszczenia do egzaminu jest uzyskanie zaliczenia.</p>
<p>Treści modułu kształcenia</p>	<ol style="list-style-type: none"> 1. Wprowadzenie do kompilacji 2. Analiza leksykalna 3. Generatory analizatorów leksykalnych LEX, FLEX: opis działania i ich wykorzystanie. 4. Analiza składniowa - gramatyki SLR(1), LALR(1), LR(1) i ich analizatory 5. Generatory redukcyjnych analizatorów składniowych typu LALR(1) - YACC, BISON i ich wykorzystanie 6. Analiza semantyczna sterowana składnią <ol style="list-style-type: none"> a. Statyczna analiza semantyczna dla wybranych konstrukcji językowych.

METODY KONSTRUKCJI KOMPILATORÓW
Compiler Construction Methods
Rok 2014/2015 Semestr zimowy

	<ul style="list-style-type: none"> b. Organizacja tablicy symboli dla języków typu proceduralnego, c. Generacja kodu pośredniego programu 7. Struktury danych wykorzystywane podczas generacji kodu, 8. Organizacja pamięci w czasie wykonania programu 9. Zarządzanie pamięcią obiektów i algorytmy odzyskiwania zaśmieconej pamięci. 10. Optymalizacja kompilatorów. <ul style="list-style-type: none"> a. metody optymalizacji lokalnej kodu pośredniego b. metody optymalizacji globalnej kodu pośredniego 11. Generowanie kodu wynikowego
Wykaz literatury podstawowej i uzupełniającej, obowiązującej do zaliczenia danego modułu	<ol style="list-style-type: none"> 1. A.V. Aho, R. Sethi, J.D. Ullman, „Kompilatory – reguły, metody i narzędzia”, WNT Warszawa 2002 2. W.M.Waite, G. Goos , „ Konstrukcja kompilatorów”. WNT Warszawa 1989. 3. D. Gries , „ Konstrukcja kompilatorów dla maszyn cyfrowych”, WNT Warszawa 1989. 4. Dokumentacja UNIX-a dla LEX-a, FLEX-a i YACC-a.
Metody i kryteria oceniania	<p>Student jest oceniany na podstawie punktów uzyskiwanych z samodzielnie implementowanych modułów kompilatorów, zadań tablicowych, kolokwiów i egzaminu ustnego. Skalę ocen ustala wykładowca.</p>
Wymiar, zasady i forma odbywania praktyk, w przypadku, gdy program kształcenia przewiduje praktyki	<p>Nie dotyczy</p>