



Karta przedmiotu

| | | | | | | | |
|---|---|--|--|-----------------------|---------|------------|-------|
| Nazwa i kod przedmiotu | Programming languages, PG_00045303 | | | | | | |
| Kierunek studiów | Inżynieria danych | | | | | | |
| Data rozpoczęcia studiów | październik 2020 r. | Rok akademicki realizacji przedmiotu | 2021/2022 | | | | |
| Poziom kształcenia | I stopnia - inżynierskie | Grupa zajęć | Grupa zajęć obowiązkowych z zakresu kierunku studiów Grupa zajęć powiązanych z prowadzonymi badaniami naukowymi w dziedzinie nauki związanej z kierunkiem - profil ogólnoakademicki | | | | |
| Forma studiów | stacjonarne | Sposób realizacji | na uczelni | | | | |
| Rok studiów | 2 | Język wykładowy | angielski | | | | |
| Semestr studiów | 3 | Liczba punktów ECTS | 2.0 | | | | |
| Profil kształcenia | ogólnoakademicki | Forma zaliczenia | zaliczenie | | | | |
| Jednostka prowadząca | Wydział Elektroniki -> Telekomunikacji i Informatyki -> Katedra Algorytmów i Modelowania Systemów | | | | | | |
| Imię i nazwisko wykładowcy (wykładowców) | Odpowiedzialny za przedmiot | dr inż. Piotr Mironowicz | | | | | |
| | Prowadzący zajęcia z przedmiotu | mgr inż. Tomasz Goluch dr hab. inż. Jan Daciuk dr inż. Piotr Mironowicz dr inż. Jerzy Dembski | | | | | |
| Formy zajęć i metody nauczania | Forma zajęć | Wykład | Ćwiczenia | Laboratorium | Projekt | Seminarium | RAZEM |
| | Liczba godzin zajęć | 15.0 | 0.0 | 15.0 | 0.0 | 0.0 | 30 |
| W tym liczba godzin zajęć na odległość: 0.0 | | | | | | | |
| Aktywność studenta i liczba godzin pracy | Aktywność studenta | Udział w zajęciach dydaktycznych, objętych planem studiów | Udział w konsultacjach | Praca własna studenta | RAZEM | | |
| | Liczba godzin pracy studenta | 30 | 4.0 | 16.0 | 50 | | |
| Cel przedmiotu | Celem przedmiotu jest zapoznanie się z popularnymi paradygmatami programowania i zdobycie umiejętności ich praktycznego wykorzystywania. | | | | | | |
| Efekty uczenia się przedmiotu | Efekt kierunkowy | Efekt z przedmiotu | Sposób weryfikacji i oceny efektu | | | | |
| | [K6_W05] zna i rozumie modele programowania i ewolucję związanych z nimi języków. zna metody analizowania i projektowania systemów informatycznych i wykorzystywane w nich języki modelowania, a także podstawowe platformy programowania obiektowego | Student zna koncepcyjnie i historycznie istotne języki programowania, takie jak Modula, Ada, Smalltalk, Haskell, Prolog. | [SW1] Ocena wiedzy faktograficznej | | | | |
| | [K6_U01] programuje w językach programowania proceduralnego, obiektowego, funkcjonalnego i w logice, koduje programy na poziomie instrukcji procesora, uruchamia i testuje programy | Student umie programować w językach paradygmatu proceduralnego i obiektowego (Modula, Smalltalk), funkcyjnego (Haskell) i w logice (Prolog). | [SU4] Ocena umiejętności korzystania z metod i narzędzi [SU1] Ocena realizacji zadania | | | | |

Treści przedmiotu

1. Programowanie proceduralne.
2. Składnia liniowa. FORTRAN
3. Rekordy aktywacyjne i podprogramy
4. Rekurencyjne wywołanie procedur.
5. Składnia blokowa. Abstrakcja przepływu sterowania.
6. Wiązanie nazw z obiektami. Zasięg wiązania.
7. Parametry wywołania procedury. Tryby przekazu parametrów.
8. Rekordy aktywacyjne dla języków z rekurencją.
9. Wywołania statyczne i dynamiczne. ALGOL. PASCAL.
10. Ograniczenia języków blokowych.
11. Abstrakcja danych i ochrona dostępu.
12. Modularyzacja. MODULA-2. ADA83, ADA95
13. Sytuacje wyjątkowe. Modele obsługi wyjątków.
14. Współprogramy i procedury współbieżne. Rendezvous.
15. Programowanie obiektowe. Obiekty, klasy, hierarchie.
16. Typy dynamiczne. Polimorfizm. SMALLTALK. C++.
17. Rekurencyjne interpretowanie poleceń.
18. Przekształcenia symboliczne. Rekurencja ogonowa.
19. Funkcyjny paradygmat programowania.
20. Haskell. Podstawowe konstrukcje i operacje, tworzenie modułów, wydajność.
21. Tacit programming.
22. LISP. Ogólny zarys.
23. Prolog jako przykład programowania w logice.
24. Definiowanie języków. Klasyfikacja Chomsky'ego, kompilatory.
25. Monady.

| | | | |
|---|-----------------------------|--|-------------------------|
| Wymagania wstępne i dodatkowe | | | |
| Sposoby i kryteria oceniania osiągniętych efektów uczenia się | Sposób oceniania (składowe) | Próg zaliczeniowy | Składowa oceny końcowej |
| | kolokwia | 50.0% | 40.0% |
| | projekt | 50.0% | 60.0% |
| Zalecana lista lektur | Podstawowa lista lektur | <p>1. S. Mangano: XSLT receptury, wyd.2, Helion 2007</p> <p>2. Cincom Smalltalk Downloads, http://www.cincomsmalltalk.com/</p> <p>3. SAXON - The XSLT and XQuery Processor, http://saxon.sourceforge.net/</p> <p>4. W.F. Clocksin, W.F., Mellish, C.S.: Prolog Programowanie. Helion 2003</p> <p>5. Ada Programming, http://en.wikibooks.org/wiki/Ada</p> <p>6. SWI-Prolog downloads, www.swi-prolog.org/download.html</p> <p>7. ADA Core, the GNAT Pro Company, http://www.adacore.com/home, https://libre.adacore.com/</p> <p>8. D. S. Touretzky: Common Lisp: A Gentle Introduction to Symbolic Computation, http://www.cs.cmu.edu/~dst/LispBook/</p> <p>9. Z. Huzar, Z. Fryźlewicz, I. Dubielewicz, B. Hnatk: Ada 95, Helion 1998</p> <p>10. Polski serwis języka Smalltalk, http://www.objectspace.net/</p> | |
| | Uzupełniająca lista lektur | http://en.wikipedia.org/wiki/Programming_paradigm | |
| | Adresy eZasobów | | |
| Przykładowe zagadnienia/ przykładowe pytania/ realizowane zadania | | | |
| Praktyki zawodowe w ramach przedmiotu | Nie dotyczy | | |