

JĘZYKI PROGRAMOWANIA

Mechanika i Budowa Maszyn – SS i SN II^o (ostatnia aktualizacja 23 września 2025)

Wykłady¹ (15 h SS / 8 h SN) – prowadzący: Prof. dr hab. inż. Wojciech Sobieski:

- **Generacje języków programowania:** historia komputerów, generacje komputerów, prawo Moora, ewolucja języków programowania, generacje języków programowania, paradygmaty programowania.
- **Wprowadzenie do programowania:** język, język programowania, translator, kompilator, asembler, interpreter, kod źródłowy, kod maszynowy, pliki dołączane, biblioteka, biblioteka statyczna, biblioteka dynamiczna, etapy kompilacji, preprocesing, kompilacja, konsolidacja, środowisko programistyczne, klasyfikacja i przykłady języków programowania.
- **Kompilatory Fortranu:** język Fortran, standardy języka Fortran, kompilatory języka Fortran, sposoby edycji kodu źródłowego, zmienne środowiskowe, polecenie kompilacji, uruchamianie programu.
- **Elementy języków programowania:** znaki i symbole, słowa kluczowe, struktura kodu, zasady edycji kodu, identyfikatory, operacje wejścia-wyjścia, typy zmiennych, formatowanie, zakresy obowiązywania zmiennych, działania na plikach, komunikacja z systemem operacyjnym, zarządzanie pamięcią, zmienne i stałe, wyjątki, wyrażenia arytmetyczne, logiczne i tekstowe, operatory arytmetyczne, logiczne, relacji i tekstowe,
- **Instrukcje języków programowania:** instrukcja, instrukcje niskiego i wysokiego poziomu, instrukcje czynne (przypisania, skoku, zatrzymania, wstrzymania, powrotu, końca, warunkowa, wyboru, powtórzeń (rodzaje instrukcji powtórzeń), wejścia-wyjścia, obsługi wyjątków), instrukcje bierne (specyfikacje segmentów kodu, procedura, funkcja, cechy procedury i funkcji, specyfikacje formatu danych, inicjowanie danych).
- **Algorytmy:** algorytm, cechy algorytmu, iteracja, rekursja, obszary stosowania algorytmów, zapis algorytmów, diagramy, zasady tworzenia diagramów, etapy tworzenia algorytmów, elementy algorytmów, rodzaje błędów w programach komputerowych (językowe, semantyczne, logiczne, algorytmiczne), kryteria optymalizacji programów komputerowych, przykłady sposobów optymalizacji programów komputerowych.
- **Dobre praktyki:** przegląd zasad i wskazówek poprawiających jakość kodów źródłowych.

Ćwiczenia² (30 h SS / 16 h SN) – prowadzący: Prof. dr hab. inż. Wojciech Sobieski:

1. **Nauka elementów programowania imperatywnego w języku FORTRAN (Edi / gfortran):**
 - struktura kodu źródłowego (PROGRAM, END), podstawowe operacje wejścia-wyjścia (PRINT, WRITE, READ), instrukcja przypisania (=), zmienne proste i indeksowane (INTEGER, REAL, COMPLEX, LOGICAL, CHARACTER, TRUE, FALSE, LEN), zmienne statyczne i dynamiczne (ALLOCATABLE, ALLOCATE, DEALLOCATE), stałe (PARAMETER), jawna i niejawną deklaracja typu (IMPLICIT NONE), formatowanie danych (FORMAT);
 - wyrażenia i operatory arytmetyczne (+, -, *, /, **), wyrażenia i operatory logiczne (.NOT., .AND., .OR., .EQV., .NEQV.), operatory relacji (.GT., .GE., .LT., .LE., .EQ., .NE.), wyrażenia i operatory tekstowe (TRIM, //, :), instrukcje sterujące (GOTO, IF, THEN, ELSE, SELECT CASE, DO, DO WHILE, CONTINUE);
 - funkcje lokalne, wewnętrzne i zewnętrzne (FUNCTION), procedury (SUBROUTINE), pliki dołączane (INCLUDE), typy danych.
2. **Ćwiczenia z języka FORTRAN:**
 - tworzenie przykładowych programów obliczeniowych.
3. **Nauka podstaw wizualizacji danych:**
 - podstawy wizualizacji 2D w trybie skryptowym w środowisku graficznym Gnuplot;
 - podstawy wizualizacji 3D z wykorzystaniem plików VTK i programu ParaView.
4. **Nauka elementów programowania wizualnego w języku Object Pascal (Free Pascal / Lazarus):**
 - tworzenie nakładek graficznych na wybrane programy obliczeniowe.

Uwagi:

- Zakres wykładów i ćwiczeń może ulec niewielkiej zmianie.
- Tematy z programowania wizualnego są opcjonalne.
- Nauka programowania odbywać się będzie w systemach operacyjnych Windows oraz Unix/Linux.
- Zabrania się rejestracji wykładów i ćwiczeń w formie nagrań wideo lub audio.

Zasady zaliczenia przedmiotu (przedmiot kończy się egzaminem):

- Obecność na wykładach i ćwiczeniach.
- Ocena pozytywna z pracy na zajęciach.
- Rozliczenie się z zadań realizowanych na zajęciach.
- Zaliczenie egzaminu obejmującego zagadnienia **teoretyczne** oraz **praktyczne**.

Literatura:

1. Sobieski W., GNU Fortran z elementami wizualizacji danych, wyd. UWM, 2008.
2. The gfortran team, Using GNU Fortran [on-line] URL: <https://gcc.gnu.org/onlinedocs/gcc-11.2.0/gfortran.pdf>, wyd. GCC, 2021.
3. Williams T., Kelley C., Gnuplot - An Interactive Plotting Program [on-line] URL: http://www.gnuplot.info/docs_5.4/Gnuplot_5_4.pdf wyd. Gnuplot, 2020.
4. File Format for VTK [on-line] URL: <https://vtk.org/wp-content/uploads/2015/04/file-formats.pdf>.
5. van Canneyt M., Free Pascal Reference guide [on-line] URL: <https://ftp.icm.edu.pl/packages/fpc/docs-pdf/ref.pdf>, wyd. Lazarus, 2021.
6. Trykozko A., Ćwiczenia z języka Fortran, wyd. MIKOM, 1999.
7. Chrobak D., FORTRAN – Praktyka programowania, wyd. MIKOM, 2003.
8. Piechna J. R.: Programowanie w języku Fortran 90 i 95. Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa 2000.
9. Sobieski W., Edytor Edi/Hugo – podręcznik użytkownika [on-line], wyd. KMiPKM, 2005.

¹ Na studiach niestacjonarnych realizowane są skrócone wersje wykładów.

² Na studiach niestacjonarnych realizowanych jest mniej przykładów.