Nazwa przedmiotu						
Metody algebry komputerowej w technice Computer algebra methods in technology						
Dyscyplina					Oznaczenie przedmiotu	
Informatyka					2S_IS1_MAKwT	
Rodzaj przedmiotu	Stopień studiów	Tryb studiów	Język zajęć		Rok	Semestr
Do wyboru	1	stacjonarne	polski		III	V
Rodzaj zajęć		Wyk.	Ćw.	Lab.	Sem.	Proj.	Liczba punktów ECTS
Liczba godzin w semestrze		30	0	15	0	0	4
Koordynator	Dr hab. inż. Andrey Grishkevich, prof. PCz. a.grischkevich@el.pcz.czest.pl						
Prowadzący	Dr hab. inż. Andrey Grishkevich, prof. PCz. a.grischkevich@el.pcz.czest.pl Prof. dr hab. Krzysztof Sokalski sokalski@el.pcz.czest.pl						

I. KARTA PRZEDMIOTU

Cel przedmiotu

C1. Opanowanie podstaw programowania symbolicznego.
C2. Opanowanie podstaw wykonywania obliczeń za pomocą algebry komputerowej.
C3. Opanowanie metod rozwiązywania problemów matematycznych za pomocą algebry komputerowej.
C4. Opanowanie metod grafiki i animacji w algebrze komputerowej w zakresie tworzenia wykresów.
C5. Rozwiązanie wybranych problemów technicznych za pomocą algebry komputerowej.

Wymagania wstępne w zakresie wiedzy, umiejętności i innych kompetencji

1. Programowanie w jednym z następujących języków: C++, Java, Pascal, VB.net.
2. Matematyka dyskretna i algebra liniowa.
3. Analiza matematyczna.

Efekty kształcenia

EK1. Student zna i rozumie składnię i semantykę języka symbolicznego oraz strukturę danych.
EK2. Student potrafi zaprogramować w języku symbolicznym następujące zagadnienia: wyrażenia matematyczne, listy i zbiory, równania i układy równań algebraicznych, równania i układy równań różniczkowych zwyczajnych, równania różniczkowe cząstkowe, całki nieoznaczone i całki oznaczone.
EK3. Student potrafi tworzyć w języku symbolicznym procedury i funkcje, stosować operatory, pętle oraz instrukcje logiczne. Potrafi wykonywać operacje na listach i zbiorach. Student potrafi używać instrukcje wejścia i wyjścia oraz eksportować wyniki w różnych formatach.

Treści programowe: wykłady	Liczba godzin
W 1 – Typy zmiennych i operatorów, wykonywanie operacji arytmetycznych, stosowanie bibliotecznych funkcji i procedur.	2
W 2 – Budowa wyrażeń i zdań. Operacje na wyrażeniach i zdaniach. Tworzenie list, zbiorów i tabel. Operacje na listach, zbiorach i zdaniach.	2
W 3 – Tworzenie i operacje na kolejkach i stosach. Tworzenie własnych funkcji i procedur. Operatory pochodnej i całki.	2
W 4 – Konwersja pomiędzy różnymi typami zmiennych i struktur. Instrukcje sterujące, instrukcje I/O, polecenia zarządzające plikami.	2
W 5 – Interakcyjne tworzenie grafiki i animacji .	2
W 6 -. Tworzenie kodu grafiki w oddzielnym pliku.	2
W 7 – Tworzenie grafiki 3d.	2
W 8 – Debugowanie i obsługa błędów. Wydajność obliczeniowa .	2
W 9 – Tworzenie procedur zwracających procedury.	2
W 10 – Tworzenie modułów wbudowanych: Składnia i semantyka.	2
W 11 – Moduły c.d.: pakiety procedur	2
W 12 – Obiekty. Programowanie zorientowane obiektowo.	2
W 13 – Project1. Symboliczny operator różniczkowania.	2
W 14 – Projekt2. Wyszukiwarka tekstu.	2
W 15 – Projekt2. Wyszukiwarka tekstu (c.d.)	2
SUMA	**30**

Treści programowe: laboratorium	Liczba godzin
L 1 – Organizacja warsztatu pracy	1

L 2 – Ćwiczenia z zakresu: typów zmiennych i operatorów, wykonywania operacji arytmetycznych, stosowania bibliotecznych procedur, budowy wyrażeń i zdań.	2
L 3 – Ćwiczenia z zakresu: operacji na wyrażeniach i zdaniach, tworzenia list , zbiorów i tabel, operacji na listach, zbiorach i tabelach. Konwersji pomiędzy różnymi typami zmiennych i struktur, oraz tworzenie i operacje na kolejkach i stosach.	2
L 4 – Ćwiczenia z zakresu tworzenia własnych procedur i funkcji, stosowania instrukcji sterujących oraz instrukcji I/O, debugowania i wydajności.	2
L 5 – Rozwiązywanie równań sprzężonych obwodów liniowych RLC	2
L 6 – Rozwiązywanie równań solitonowych i badanie propagacji fali elektromagnetycznej w światłowodach dielektrycznych	2
L 7 – Analiza powstawania fali uderzeniowej w wodzie przepływającej przez rurę.	2
L.8 – Odbiór projektów własnych. Zaliczenie laboratorium.	2
SUMA	**15**

Narzędzia dydaktyczne

1. Wykład z prezentacją multimedialną. Rzutnik komputerowy wraz z ekranem. Tablica klasyczna.
2. Instrukcje do ćwiczeń laboratoryjnych w postaci plików .doc, .docx, .pdf, .jpg, .txt, xls, .xlsx, .xlsm, .zip, .rar.
3. Komputery z systemem operacyjnym Windows 7/8/10 i zainstalowanymi pakietami Maple, Visual Studio, Microsoft Office 2007/2010/2013/2016 oraz przeglądarką plików .pdf, .html.
4. Podręczniki i skrypty.
5. Internet.

Sposoby oceny efektów kształcenia (F – ocena Formująca, P – ocena Podsumowująca)

F1. Aktywność na zajęciach (obecność, dyskusja, praca, wykonanie testów).
P1. Wykonanie obowiązkowego zestawu zadań (projektów) w trakcie zajęć laboratoryjnych (laboratorium).
P2. Test zaliczeniowy (wykłady).

Obciążenie pracą studenta

Forma aktywności	Średnia liczba godzin na zrealizowanie aktywności
Godziny kontaktowe z prowadzącym	45
Zapoznanie się ze wskazaną literaturą	20
Przygotowanie do zajęć	20
Przygotowanie do testu	15
Przygotowanie sprawozdań	10
Sumaryczna liczba godzin/punktów ECTS dla przedmiotu	**110 / 4 ECTS**

Wykaz literatury podstawowej i uzupełniającej

1. Artur Krowiak, Maple. Podręcznik, ISBN: 9788324633128 / 978-83-246-3312-8, Helion Gliwice.
2. http://mmc.geofisica.unam.mx/acl/edp/Ejemplitos/05-EcuacionesDiferencialesParciales/Mathematica/ABGMATHs1.pdf
3. M.B. Morgan, K.O. Geddes, K.M. Heal, G. Labahn, S.M. Vorkoetter, J.McCarron, P. De Marco. Maple Advanced Programming Guide, Maplesoft, Waterloo Maple Inc.

Macierz realizacji efektów kształcenia

Efekt kształcenia	Odniesienie efektu do efektów kształcenia dla dyscypliny naukowej Elektrotechnika*	Cele przedmiotu	Forma zajęć	Narzędzia dydaktyczne	Sposób oceny
EK1	KINF1_W01 KINF1_W03	C1	W, Lab	1,2,3,4,5	F1, P1, P2
EK2	KINF1_W09 KINF1_U07 KINF1_U08 KINF1_U17	C1, C2	W, Lab	1,2,3,4,5	F1, P1, P2
EK3	KINF1_W03 KINF1_W09 KINF1_U07 KINF1_U08 KINF1_U17	C1, C3	W, Lab	1,2,3,4,5	F1, P1, P2

* – wg załącznika

II. FORMY OCENY – SZCZEGÓŁY

Ocena	Efekty
EK1	**EK 1 – Student zna i rozumie składnię i semantykę języka symbolicznego oraz strukturę danych.**

2	Student nie potrafi napisać aplikacji w języku symbolicznym bez błędów w składni, bez błędów semantycznych i bez błędów w strukturze danych.
3	Student potrafi napisać aplikację w języku symbolicznym bez błędów w składni, bez błędów semantycznych i bez błędów w strukturze danych. Jednak wynik ten osiąga po dużej liczbie usuniętych błędów w składni lub błędów syntaktycznych lub błędów w strukturze danych.
3.5	Student rozumie komunikaty kompilatora/interpretera o błędach formalnych i potrafi je korygować posługując się debugerem.
4	Student potrafi napisać aplikację w języku symbolicznym bez błędów w składni, bez błędów semantycznych i bez błędów w strukturze danych. Wynik ten osiąga po korekcji małej liczby błędów.
4.5	Student potrafi napisać aplikację w języku symbolicznym bez błędów w składni, bez błędów semantycznych i bez błędów w strukturze danych.
5	Student potrafi weryfikować pod względem merytorycznym otrzymane wyniki, rozumie komunikaty kompilatora/interpretera o błędach formalnych i potrafi je korygować posługując się debugerem. Student posługuje się pełnym zestawem komend debugera oraz potrafi posługiwać się punktami kontrolnymi.
EK2	**EK 2 - Student potrafi zaprogramować w języku symbolicznym następujące zagadnienia: wyrażenia matematyczne, listy i zbiory, równania i układy równań algebraicznych, równania i układy równań różniczkowych zwyczajnych, równania różniczkowe cząstkowe, całki nieoznaczone i całki oznaczone.**
2	Student nie potrafi bezbłędnie zaprogramować w języku symbolicznym, chociaż jednego z następujących zagadnień: wyrażeń matematycznych, listy, zbioru, macierzy, równania algebraicznego, układu równań algebraicznych, równania różniczkowego zwyczajnego, układu równań różniczkowych zwyczajnych, równania różniczkowego cząstkowego, całki nieoznaczonej, całki oznaczonej.
3	Student potrafi bezbłędnie zaprogramować w języku symbolicznym następujące zagadnienia: wyrażenia matematyczne, listy, zbiory, macierze, równania i układy równań algebraicznych, równania i układy równań różniczkowych zwyczajnych, równania różniczkowe cząstkowe, całki nieoznaczone i oznaczone. Jednak wynik ten osiąga po dużej liczbie usuniętych błędów.
3.5	Student rozumie komunikaty kompilatora/interpretera o błędach formalnych i potrafi je korygować posługując się debugerem.
4	Student potrafi bezbłędnie zaprogramować w języku symbolicznym następujące zagadnienia: wyrażenia matematyczne, listy, zbiory, macierze, równania i układy równań algebraicznych, równania i układy równań różniczkowych zwyczajnych, równania różniczkowe cząstkowe, całki nieoznaczone i oznaczone. Wynik ten osiąga po korekcji małej liczby błędów.
4.5	Student potrafi bezbłędnie zaprogramować w języku symbolicznym następujące zagadnienia: wyrażenia matematyczne, listy, zbiory, macierze, równania i układy równań algebraicznych, równania i układy równań różniczkowych zwyczajnych, równania różniczkowe cząstkowe, całki nieoznaczone i oznaczone.
5	Student potrafi weryfikować pod względem merytorycznym otrzymane wyniki, rozumie komunikaty kompilatora/interpretera o błędach formalnych i potrafi je korygować posługując się debugerem. Student posługuje się pełnym zestawem komend debugera oraz potrafi posługiwać się punktami kontrolnymi.
EK3	**EK 3 – Student potrafi tworzyć w języku symbolicznym procedury i funkcje, stosować operatory, pętle oraz instrukcje logiczne. Potrafi wykonywać operacje na listach i zbiorach. Student potrafi używać instrukcje wejścia i wyjścia oraz eksportować wyniki w różnych formatach.**
2	Student nie potrafi bezbłędnie tworzyć w języku symbolicznym, chociaż jednego z następujących zagadnień: procedury, funkcji, stosować operatorów, stosować pętli, stosować instrukcji logicznych. Lub nie potrafi wykonywać operacji na listach lub nie potrafi wykonywać operacji na zbiorach. Lub student nie potrafi używać instrukcji wejścia lub instrukcji wyjścia lub nie potrafi eksportować wyników obliczeń.
3	Student potrafi bezbłędnie tworzyć w języku symbolicznym procedury i funkcje, stosować operatory, pętle oraz instrukcje logiczne. Potrafi wykonywać operacje na listach i zbiorach. Student potrafi używać instrukcje wejścia i wyjścia oraz eksportować wyniki w różnych formatach. Jednak wynik ten osiąga po dużej liczbie usuniętych błędów.
3.5	Student rozumie komunikaty kompilatora/interpretera o błędach formalnych i potrafi je korygować posługując się debugerem.
4	Student potrafi bezbłędnie tworzyć w języku symbolicznym procedury i funkcje, stosować operatory, pętle oraz instrukcje logiczne. Potrafi wykonywać operacje na listach i zbiorach. Student potrafi używać instrukcje wejścia i wyjścia oraz eksportować wyniki w różnych formatach. Wynik ten osiąga po korekcji małej liczby błędów.
4.5	Student potrafi bezbłędnie tworzyć w języku symbolicznym procedury i funkcje, stosować operatory, pętle oraz instrukcje logiczne. Potrafi wykonywać operacje na listach i zbiorach. Student potrafi używać instrukcje wejścia i wyjścia oraz eksportować wyniki w różnych formatach.
5	Student potrafi weryfikować pod względem merytorycznym otrzymane wyniki, rozumie komunikaty kompilatora/interpretera o błędach formalnych i potrafi je korygować posługując się debugerem. Student posługuje się pełnym zestawem komend debugera oraz potrafi posługiwać się punktami kontrolnymi.

III. INNE PRZYDATNE INFORMACJE O PRZEDMIOCIE

1. Wszelkie informacje dla studentów na temat planu zajęć dostępne są na tablicy ogłoszeń oraz na stronie el.pcz.pl.
2. Informacje na temat warunków zaliczania zajęć przekazywana jest studentom podczas pierwszych zajęć.
3. Instrukcje do zajęć laboratoryjnych i treści wykładów będą umieszczane na wskazany przez prowadzącego adres poczty elektronicznej. Przejrzenie instrukcji wymaga zainstalowania oprogramowania czytającego/interpretacyjącego pliki .doc, .docx, .pdf, .jpg, .txt, xls, .xlsx, .xlsm, .zip, .rar.
4. Zajęcia laboratoryjne będą odbywać się w sali IK2 Wydziału Elektrycznego lub równoważnej.