Nazwa przedmiotu							
Algorytmy i złożoności Algorithms and complexities							
Dyscyplina							Oznaczenie przedmiotu
Informatyka							3K_INS1_AiZ
Rodzaj przedmiotu	Stopień studiów	Tryb studiów		Język zajęć		Rok	Semestr
obowiązkowy	1	niestacjonarne		polski		II	III
Rodzaj zajęć		Wyk.	Ćw.	Lab.	Sem.	Proj.	Liczba punktów ECTS
Liczba godzin w semestrze		9	0	18	0	0	6
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 Dr hab. inż. Sławomir Gryś, prof. PCz. grys@el.pcz.czest.pl Dr Marek Matusiewicz mm@el.pcz.czest.pl						

I. KARTA PRZEDMIOTU

Cel przedmiotu

C1. Znajomość złożoności obliczeniowej algorytmów i umiejętność porównywania algorytmów.
C2. Tworzenie prostych programów komputerowych w języku C++.
C3. Pisanie rekurencyjnych programów komputerowych.
C4. Znajomość algorytmów sortowania.
C5. Umiejętność wykorzystania struktury danych: lista, stos, drzewo, graf.
C6. Wiedza i umiejętności wykorzystania w praktyce algorytmów na grafach: najkrótsze ścieżki i minimalne drzewo rozpinające.

Wymagania wstępne w zakresie wiedzy, umiejętności i innych kompetencji

1. Wiedza z zakresu matematyki: rekurencja, ciąg Fibonacciego, funkcji (exp(x), ln(x), Pn(x), Π, Σ), granica ciągu (lim), rzędy wielkości i porównywanie funkcji, podstawy kombinatoryki.
2. Wiedza z zakresu podstaw programowania: algorytm, kroki algorytmu, instrukcji sterujących, instrukcji warunkowych, instrukcji pętli, typy danych, translator, program komputerowy.
3. Umiejętność obsługi komputera i pracy z INTERNET.
4. Znajomość języka angielskiego w stopniu wystarczającym do czytania komunikatów translatora.

Efekty kształcenia

EK1. Student potrafi stworzyć program rekurencyjny
EK2. Student zna algorytmy sortowania
EK3. Student zna struktury danych

Treści programowe: wykłady	Liczba godzin
W 1 – Algorytm. Podstawy analizy algorytmów. Techniki projektowania algorytmów: dziel i rządź, programowanie dynamiczne, algorytmy zachłanne, przeszukiwanie z nawrotami, heurystyki. – C++. Visual Studio. Pierwszy program. Analiza programów. Komunikowanie z użytkownikiem. Identyfikatory. Wyrażenia i operacje. Operacje arytmetyczne. Operatory relacji. Deklaracje typów. Tablice. Instrukcje sterujące. Funkcje. Wskaźniki. Klasy. Konstruktorzy.	1
W 2 – Rekurencja. Silnia. Ciąg Fibonacciego. Jak wykonują się programy rekurencyjnie? Niebezpieczeństwa rekurencji (stack overflow, stąd do wieczności). Typy programów rekurencyjnych. Uwagi praktyczne na temat technik rekurencyjnych.	1
W 3 – Algorytmy sortowania. Sortowanie przez wstawianie. Sortowanie bąbelkowe. Quicksort. Uwagi praktyczne.	1
W 4 – Kolejki priorytetowe. Podstawowe operacje na kolejkach priorytetowych. Sterta. Realizacja sterty. Zastosowanie sterty do sortowania danych.	1
W 5 – Analiza sprawności algorytmów. Złożoność obliczeniowa (czasowa, pamięciowa, pesymistyczną, gwarantowana, oczekiwana, minimalna, przykłady) algorytmów. Rzędy wielkości i porównywanie funkcji, notacja O i o. Funkcje złożoności (logarytmiczna, liniowa, liniowo-logarytmiczna, kwadratowa, wielomianowa, wykładnicza). Złożoność algorytmów sortowania. Problemy obliczeniowo trudne: NP-zupełność, nierozstrzygalność. Wnioski.	1
W 6 – Abstrakcyjne struktury danych. Listy. Lista jednokierunkowa. Realizacja struktur danych listy jednokierunkowej. Lista dwukierunkowa. Lista cykliczna. Stos. Zasada działania stosu. Biblioteki standartowe <list>, <stack>.	1
W 7 – Drzewa i ich reprezentacje. Drzewa binarne. Drzewa binarne i wyrażenia arytmetyczne. Postać wrostkowa, postfiksowa (odwrotna notacja Polska), prefiksowa.	1
W 8 – Uniwersalna struktura słownikowa.	1

W 9 – Grafy. Podstawowe pojęcia teorii grafów. Grafy i sieci. Grafy i łamigłówki, współczesne rozwiązania komputerowe łamigłówek. Zastosowania grafów. Problemy i algorytmy grafowe. Prezentacje grafów. Algorytmy przechodzenia grafu. Metoda oznaczeń. Przeszukiwanie grafu w głąb. Przeszukiwanie grafu wszerz. Złożoność algorytmów przeszukiwania. Algorytm szukania komponent spójności. Drzewa grafów. Właściwości drzew. Minimalne (maksymalne) drzewo rozpinające. Zastosowanie minimalnych (maksymalnych) drzew rozpinających. Algorytm (Kruskala, Prima) znajdowania minimalnego (maksymalnego) drzewa rozpinającego sieci w postaci: a) listy krawędzi, b) macierzy wag. Ćwiczenie algorytmu. Drogi (ścieżki) w grafie. Długość ścieżki. Zastosowanie minimalnych ścieżek. Algorytm Dijkstry znajdowania najkrótszej drogi. Metoda oznaczeń Dijkstry. Ćwiczenie algorytmu. Złożoność algorytmu Dijkstry. Test zaliczeniowy.	1
SUMA	**9**

Treści programowe: laboratorium	Liczba godzin
L 1 – Visual Studio. C++. Pierwszy program. Instrukcji sterujące. Pętli.	1
L 2 – Rekurencja. Funkcja silnia. Ciąg Fibonacciego.	2
L 3 – Algorytmy sortowania. Sortowanie przez wstawianie. Sortowanie bąbelkowe. Quicksort.	2
L 4 – Kolejki priorytetowe. Sterta. Zastosowanie sterty do sortowania danych.	2
L 5 – Test zaliczeniowy (C++, rekurencja, sortowanie).	1
L 6 – Struktury danych. Lista (jednokierunkowa, dwukierunkowa, cykliczna). Stos.	3
L 7 – Drzewa. Wyrażenia arytmetyczne.	1
L 8 – Uniwersalna struktura słownikowa.	1
L 9 – Grafy. Reprezentacji grafów. Minimalne drzewo rozpinające. Maksymalne drzewo rozpinające.	2
L 10 – Najkrótsze ścieżki.	2
L 11 – Odbiór obowiązkowego zestawu zadań. (Test zaliczeniowy.)	1
SUMA	**18**

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, .cpp, .exe.
3. Internet.
3. Komputery z systemem operacyjnym Windows 7/8/10 i zainstalowanym środowiskiem programowania Microsoft Visual Studio 2015/2017, pakietem Microsoft Office 2007/2010/2013/2016 oraz przeglądarką plików .pdf, .html.
4. Podręczniki i skrypty.

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

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

Obciążenie pracą studenta

Forma aktywności	Średnia liczba godzin na zrealizowanie aktywności
Godziny kontaktowe z prowadzącym	27
Zapoznanie się ze wskazaną literaturą	43
Przygotowanie do zajęć	50
Przygotowanie do testu	20
Przygotowanie sprawozdań	20
Sumaryczna liczba godzin/punktów ECTS dla przedmiotu	**160 / 6 ECTS**

Wykaz literatury podstawowej i uzupełniającej

1. Wróblewski P.: Algorytmy. Struktury danych i techniki programowania. Wydawnictwo Helion, Gliwice.
2. Grebosz J.: Symfonia C++ standard. Programirowanie w języku C++ orientowane obiektowo. Edition, Kraków.
3. Cormen T. H., Leiserson C. E., Rivest R. L., Stein C.: Wprowadzenie do algorytmów. WNT, Warszawa.
4. Banachowski L., Diks K., Rytter W.: Algorytmy i struktury danych. WNT, Warszawa.
5. Lipski W. : Kombinatoryka dla programistów. WNT, Warszawa.

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_W03 KINF1_W05 KINF1_W07 KINF1_U09	C1, C2, C3	W, Lab	1,2,3,4	F1, P1, P2

EK2	KINF1_W03 KINF1_W05 KINF1_U09	C1, C2, C4	W, Lab	1,2,3,4	F1, P1, P2
EK3	KINF1_W03 KINF1_W06 KINF1_U09	C1, C2, C5, C6	W, Lab	1,2,3,4	F1, P1, P2

* – wg załącznika

II. FORMY OCENY – SZCZEGÓŁY

Ocena	Efekty
EK1	**Student potrafi stworzyć program rekurencyjny**
2	Student nie potrafi stworzyć programu rekurencyjnego.
3	Student potrafi uruchomić programy rekurencyjny do obliczenia silni i ciąga Fibonacciego w środowisku programowania Visual Studio w języku C++.
3.5	Student zna dwa typy programów rekurencyjnych („naturalna", „z parametrem dodatkowym").
4	Student potrafi zmodyfikować program rekurencyjny.
4.5	Student potrafi niezależnie stworzyć program, która metodą rekurencji wypisuje na ekranie daną liczbę w zapisie binarnym.
5	Student potrafi użyć technik rekurencyjnych do rozwiązywania problemów osobistych (na przykład, obejścia drzewa binarnego).
EK2	**Student zna algorytmy sortowania**
2	Student nie zna algorytmów sortowania.
3	Student zna algorytmy sortowania przez wstawianie, bąbelkowe, quicksort, przy użyciu sterty.
3.5	Student potrafi ocenić złożoność algorytmów sortowania.
4	Student zna modyfikacje w/w algorytmów sortowania.
4.5	Student potrafi uruchomić programy: do sortowanie przez wstawianie, sortowania bąbelkowego, quicksort, sortowania na podstawie sterty w środowisku programowania Visual Studio w języku C++.
5	Student potrafi zmodyfikować programy do sortowania.
EK3	**Student zna struktury danych**
2	Student nie zna struktur danych.
3	Student zna struktury danych: lista, stos, drzewo, graf i ich implementacje na podstawie bibliotek standardowych.
3.5	Student potrafi zaimplementować: listę (jednokierunkową, dwukierunkową, cykliczną), stos, drzewo binarne w języku C++ w środowisku programowania Visual Studio.
4	Student potrafi dodać funkcje niestandardowe do obsługi listów i stosu.
4.5	Student potrafi zastosować drzewa do reprezentowania wyrażeń arytmetycznych i słowników.
5	Student potrafi osobiście stworzyć różne algorytmy przechodzenia drzewa (grafa) i zrealizować je w języku C++ w środowisku programowania Visual Studio.

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, .cpp, .exe.
4. Zajęcia laboratoryjne będą odbywać się w sali IK2 Wydziału Elektrycznego lub równoważnej.