

| | | | | |
|---|--------------------|--|--|--------------------|
| Nazwa przedmiotu <i>Algorytmy i struktury danych 1</i> | | Kod ECTS | | |
| Nazwa jednostki prowadzącej przedmiot <i>Uniwersytet Opolski, Wydział Matematyki, Fizyki i Informatyki, Instytut matematyki i Informatyki</i> | | | | |
| Studia | | | | |
| | Kierunek | stopień | tryb | specjalność |
| | <i>Informatyka</i> | <i>Pierwszy</i> | <i>Stacjonarne Niestacjonarne*</i> | |
| Nazwisko osoby prowadzącej (osób prowadzących) dr hab. Wiesław Szwał, prof. UO, dr Andrzej Jasiński | | | | |
| Formy zajęć, sposób ich realizacji i przypisana im liczba godzin | | Liczba punktów ECTS: 5 | | |
| A. Formy zajęć <ul style="list-style-type: none"> wykład (W), laboratorium (L). | | <i>Bilans nakładu pracy przeciętnego studenta:</i> <ul style="list-style-type: none"> 15×2 godz. = 30 godz. – udział w wykładach [^{*)} 18]; 15×2 godz. = 30 godz. – udział w laboratoriach [^{*)} 18]; 15×1 godz. = 15 godz. – analiza i przyswojenie treści poznanych na wykładach [^{*)} 27]; 5 × 1 godz. = 5 godz. – udział w konsultacjach [^{*)} 2]; 15×2 godz. = 30 godz. – przygotowanie do laboratoriów (przygotowanie założeń do zadań projektowo-programistycznych ogłaszanych po wykładach, korzystanie z literatury, korzystanie z innych źródeł) [^{*)} 42]; 12+3 godz. = 15 godz. - przygotowanie do egzaminu i obecność na egzaminie [^{*)} 18]; | | |
| B. Sposób realizacji <ul style="list-style-type: none"> zajęcia w sali wykładowej/laboratoryjnej | | Łączny nakład pracy studenta: 125 godzin, co odpowiada 5 pkt. ECTS | | |
| C. Liczba godzin wykład – 30 godzin laboratorium – 30 godzin *) <i>Studia niestacjonarne:</i> wykład – 18 godz. (2T+16Z); laboratorium – 18 godz. | | <i>w tym</i> <ul style="list-style-type: none"> nakład pracy związany z zajęciami wymagającymi bezpośredniego udziału nauczycieli akademickich: 30+30+5=65 godz., co odpowiada 2,5 pkt. ECTS; nakład pracy związany z zajęciami o charakterze praktycznym: 30+15+5+30 = 80 godz., co odpowiada 3 pkt. ECTS *) <i>na studiach niestacjonarnych:</i> <ul style="list-style-type: none"> nakład pracy związany z zajęciami wymagającymi bezpośredniego udziału nauczycieli akademickich: 38 godz., co odpowiada 1,5 pkt. ECTS; nakład pracy związany z zajęciami o charakterze praktycznym: 80 godz., co odpowiada 3 pkt. ECTS. | | |
| Status przedmiotu <ul style="list-style-type: none"> obowiązkowy (kanon) | | Język wykładowy <i>Polski (możliwość realizacji w języku angielskim)</i> | | |
| Metody dydaktyczne <ul style="list-style-type: none"> wykład / wykład problemowy / wykład z prezentacją multimedialną ćwiczenia laboratoryjne: samodzielne opracowanie projektów ilustrujących prezentowany materiał | | Forma i sposób zaliczenia oraz podst. kryteria oceny lub wymagania egzaminacyjne | | |
| | | A. Sposób zaliczenia <ul style="list-style-type: none"> zaliczenie z oceną (laboratorium) egzamin z oceną (wykład) | | |
| | | B. Formy zaliczenia <ul style="list-style-type: none"> (W) egzamin ustny z oceną - prezentacja wybranego algorytmu/struktury danych, analiza algorytmu oraz omówienie ewentualnej implementacji; (L) zaliczenie z oceną: ustalenie oceny zaliczeniowej na podstawie ocen cząstkowych otrzymywanych w trakcie trwania semestru za wykonanie prac zaliczeniowych. | | |
| | | C. Podstawowe kryteria <ul style="list-style-type: none"> (W) obecność na zajęciach; pozytywnie zdany egzamin, (L) uzyskanie pozytywnej oceny końcowej. | | |
| Określenie przedmiotów wprowadzających wraz z wymogami wstępnymi <i>Należy określić:</i> <p>A. Wymagania formalne: Programowanie 1, Programowanie 2, Matematyka dyskretna, Logika dla informatyków, Analiza matematyczna,</p> <p>B. Wymagania wstępne: umiejętność programowania imperatywnego, znajomość podstawowych pojęć logiki matematycznej (logika zdaniowa, rachunek predykatów, relacje i funkcje), analizy matematycznej (granice, asymptoty, szeregi) i matematyki dyskretnej (grafy, drzewa).</p> | | | | |
| Cele przedmiotu <i>Przegląd klasycznych problemów obliczeniowych oraz metod ich rozwiązania. Zaznajomienie słuchacza z podstawowymi technikami wykorzystywanymi przy projektowaniu i analizie algorytmów .</i> | | | | |

Treści programowe

A. Problematyka wykładu

Podstawowe pojęcia: algorytm, zadanie, problem i jego specyfikacja. Algorytm jako sposób rozwiązania problemu. Pseudokod. Problem sortowania metodą porównań. Sortowanie przez wstawianie.

Podstawy analizy algorytmów – poprawność i złożoność.

Techniki projektowania algorytmów: metoda „dziel i zwyciężaj”. Sortowanie przez scalanie. Twierdzenie o rekurencji uniwersalnej.

Sortowanie przez kopcowanie. Dolne szacowanie złożoności czasowej dla problemu sortowania. Problem wyboru. Sortowanie szybkie.

Zbiory dynamiczne. Abstrakcyjne struktury danych i ich implementacje: stopy, kolejki, listy, słowniki, kolejki priorytetowe.

Abstrakcyjne struktury danych i ich implementacje: drzewa binarnych poszukiwań. Wyszukiwanie w drzewach binarnych.

Grafy, reprezentacje komputerowe grafów. Programowanie dynamiczne – problem najkrótszych ścieżek. Podstawowe algorytmy grafowe, przeszukiwanie wszerz i w głąb.

Techniki projektowania algorytmów: algorytmy zachłanne, przeszukiwanie z nawrotami.

NP-zupełność, przykłady problemów NP-zupełnych. Problem stopu.

B. Problematyka laboratorium

Sortowanie przez wstawianie, sortowanie przez scalanie, sortowanie przez kopcowanie, sortowanie szybkie - analiza teoretyczna i empiryczna. Inne algorytmy sortujące przy pomocy porównań. Stopy, kolejki, listy, kolejki priorytetowe. Drzewa binarnych poszukiwań. Reprezentacja wielotablicowa struktur ze wskaźnikami. Reprezentacje komputerowe grafów. Problem najkrótszych ścieżek. Domknięcie przechodnie grafu. Przeszukiwanie wszerz i w głąb. Kody Huffmana. Problem konika szachowego.

Wykaz literatury

A. Literatura wymagana

A.1. wykorzystywana podczas zajęć

A.2. studiowana samodzielnie przez studenta

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, Wprowadzenie do algorytmów, WNT, Warszawa 2005 (Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Wprowadzenie do algorytmów.)

2. Lech Banachowski, Krzysztof Diks, Wojciech Rytter, Algorytmy i struktury danych, WNT, Warszawa 2006..

3. Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman, Projektowanie i analiza algorytmów komputerowych, PWN, Warszawa 1983 (Projektowanie i analiza algorytmów, Algorytmy i struktury danych, Helion, 2003).

B. Literatura uzupełniająca

1. Niklaus Wirth, Algorytmy+struktury danych=programy, WNT, Warszawa 2001.

2. inne podręczniki dostępne on-line poprzez Bibliotekę Główną UO („ibuk”)

| | | | | |
|--|--|---|-------------------------|----------------------------|
| Efekty kształcenia | Wiedza | | | |
| | Symb. | Efekt | | Odniesienie |
| | W01 | Zna pojęcia: algorytm, zadanie, problem i jego specyfikacja, pseudokod, poprawność i złożoność algorytmu. | Sprawdzian pisemny | K_W02,03,04,05 |
| | W02 | Zna problem sortowania oraz standardowe algorytmy sortujące przy pomocy porównań. Zna twierdzenie o rekurencji uniwersalnej oraz dolne szacowanie złożoności czasowej dla problemu sortowania. | Zadania programistyczne | K_W02,03,05 |
| | W03 | Zna techniki projektowania algorytmów: dziel i zwyciężaj, programowanie dynamiczne, algorytmy zachłanne, przeszukiwanie z nawrotami. | | K_W03,05 |
| | W04 | Zna abstrakcyjne struktury danych: listy, drzewa, grafy, słowniki, drzewa poszukiwań binarnych, haszowanie, stopy, kolejki, kolejki priorytetowe. Zna podstawowe problemy i algorytmy na drzewach oraz grafowe. | | K_W02,03,04,05 |
| | Umiejętności: | | | |
| | Symb. | Efekt | | Odniesienie |
| | U01 | Przeprowadza analizę złożoności algorytmu oraz rozumie praktyczne znaczenie tego pojęcia. | Zadania programistyczne | K_U01,03,08,09,10,11 |
| | U02 | Konstruuje oraz implementuje algorytmy z wykorzystaniem podstawowych technik: dziel i zwyciężaj, programowanie dynamiczne, algorytmy zachłanne, przeszukiwanie z nawrotami. | | K_U01,03,04,07,08,09,10,11 |
| U03 | Stosuje wybrane struktury danych do rozwiązania określonego problemu informatycznego. Implementuje struktury dynamiczne w wybranym języku programowania. | K_U01,03,04,07,08,09,10,11 | | |
| U04 | Implementuje podstawowe algorytmy grafowe do rozwiązania problemów pokrewnych. | K_U01,03,04,07,08,09,10,11 | | |
| Kompetencje społeczne (postawy) | | | | |
| Symb. | Efekt | | Odniesienie | |
| K01 | Intuicyjnie rozumie znaczenie analizy algorytmów i dostrzega sens rozwijania swoich kompetencji w tym zakresie. Potrafi zadawać pytania zmierzające do pokonania trudności napotykanych przy rozwiązywaniu problemu. | Zadania programistyczne | K_K01,02 | |
| K02 | Potrafi zrealizować proste zadanie zespołowe, pracując w kilkusobowej grupie | Zespołowe | K_K03,09 | |

| | | | | |
|--|-----|---|-------------------------------|-------------|
| | | nad rozwiązaniem problemu algorytmicznego. Potrafi współorganizować pracę zespołu. Postępuje etycznie w zakresie wykorzystania efektów pracy innych osób. | zadanie programistyczne | |
| | K03 | Znając ograniczenia w stosowaniu metod informatyki, rozumie potrzebę popularnego przedstawiania laikom trudniejszych zagadnień oraz przedstawienia swojej opinii w tym zakresie.. | Praca pisemna lub konwersacja | K_K05,07,08 |
| | K04 | Korzysta z literatury książkowej i zasobów internetowych szukając wskazówek do rozwiązania problemu. | Zadania programistyczne | K_K06 |

Kontakt:

Wykaz numerów telefonicznych i adresów mailowych pracowników znajduje się na stronie Instytutu Matematyki i Informatyki:
www.math.uni.opole.pl