



## KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Wprowadzenie do algorytmiki

---

### Przedmiot

Kierunek studiów

Informatyka

Studia w zakresie (specjalność)

Poziom studiów

pierwszego stopnia

Forma studiów

niestacjonarne

Rok/semestr

1/1

Profil studiów

ogólnoakademicki

Język oferowanego przedmiotu

polski

Wymagalność

obieralny

---

### Liczba godzin

Wykład

12

Ćwiczenia

Laboratoria

20

Projekty/seminaria

Inne (np. online)

### Liczba punktów ECTS

4

---

### Wykładowcy

Odpowiedzialny za przedmiot/wykładowca:

mgr inż. Artur Laskowski

email: artur.laskowski@cs.put.poznan.pl

tel: 61 665 29 97

wydział: Wydział Informatyki i Telekomunikacji

adres: ul. Piotrowo 2, 60-965 Poznań

Odpowiedzialny za przedmiot/wykładowca:

---

### Wymagania wstępne

Zgodnie z podstawą programową kształcenia ogólnego dostępną na stronie: <http://bip.men.gov.pl/men>



bip/akty\_prawne/rozporządzenie\_20081223\_zal\_4.pdf zakłada się, że rozpoczynając przedmiot student ma podstawową wiedzę: - z matematyki: IV etap edukacyjny, zakres podstawowy poszerzony o rachunek różniczkowy (z zakresu rozszerzonego); - z informatyki: IV etap edukacyjny, zakres podstawowy.

Zgodnie z podstawą programową kształcenia ogólnego dostępną na stronie: [http://bip.men.gov.pl/men/bip/akty\\_prawne/rozporządzenie\\_20081223\\_zal\\_4.pdf](http://bip.men.gov.pl/men/bip/akty_prawne/rozporządzenie_20081223_zal_4.pdf) zakłada się, że rozpoczynając przedmiot student ma podstawowe umiejętności: - z matematyki: IV etap edukacyjny, zakres podstawowy poszerzony o rachunek różniczkowy (z zakresu rozszerzonego); • z informatyki: IV etap edukacyjny, zakres podstawowy.

W zakresie kompetencji społecznych student musi prezentować takie postawy jak uczciwość, odpowiedzialność, wytrwałość, ambicja i ciekawość poznawcza, kreatywność, kultura osobista, szacunek dla innych ludzi.

### Cel przedmiotu

1. Przekazanie studentom podstawowej wiedzy o programowaniu w zakresie efektywnej implementacji algorytmów w języku C++ oraz technik programistycznych usprawniających proces implementacji algorytmów oraz optymalizujących ich wydajność.
2. Przekazanie studentom wiedzy z algorytmiki w zakresie podstawowych algorytmów liczbowych, przeszukiwania binarnego, sortowania liczb, algorytmów zachłannych i programowania dynamicznego oraz kopców.
3. Przekazanie studentom wiedzy ze złożoności obliczeniowej w zakresie określania i porównywania złożoności algorytmów działających w czasie wielomianowym.
4. Rozwijanie u studentów umiejętności implementacji programistycznej poznanych algorytmów oraz struktur danych.
5. Rozwijanie u studentów umiejętności doboru odpowiedniego algorytmu i struktury danych do rozwiązywanego problemu oraz ocenę złożoności obliczeniowej i pamięciowej ich implementacji.
6. Rozwijanie u studentów umiejętności programowania zespołowego zgodnie z formułą ACM ICPC.
7. Rozwijanie u studentów umiejętności testowania zaimplementowanych algorytmów oraz ich oceny.
8. Przygotowanie studentów do startów w konkursach w programowaniu zespołowym.

### Przedmiotowe efekty uczenia się

#### Wiedza

Ma rozszerzoną i pogłębioną wiedzę z matematyki przydatną do formułowania i rozwiązywania złożonych zadań informatycznych dotyczących analizy i formalnych dowodów poprawności oraz złożoności obliczeniowej algorytmów (K1st\_W1)

Ma uporządkowaną, podbudowaną teoretycznie wiedzę ogólną w zakresie algorytmów i złożoności obliczeniowej. (K1st\_W4)

Zna podstawowe metody, techniki i narzędzia stosowane przy rozwiązywaniu prostych zadań informatycznych z zakresu analizy złożoności obliczeniowej algorytmów i problemów. (K1st\_W7)

#### Umiejętności

Potrafi wykorzystać do formułowania i rozwiązywania zadań informatycznych metody analityczne i



eksperymentalne w celu dobrania odpowiednich algorytmów i struktur danych. (K1st\_U4)  
Ma umiejętność formułowania algorytmów i ich programowania z użyciem języka C++. (K1st\_U11)  
Potrafi współdziałać i pracować w grupie, przyjmując w niej różne role w trakcie zespołowego rozwiązywania zbioru problemów. (K1st\_U18)

#### Kompetencje społeczne

Rozumie na podstawie historii standardów C++ ('11, '14, '17), że w informatyce wiedza i umiejętności bardzo szybko stają się przestarzałe. (K1st\_K1)  
Ma świadomość znaczenia wiedzy w rozwiązywaniu problemów wymagających projektowania nowych algorytmów oraz zna przykłady i rozumie przyczyny wadliwie zaimplementowanych algorytmów, które doprowadziły do poważnych strat finansowych. (K1st\_K2)

#### Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

Ocena formująca:

- a) w zakresie wykładów weryfikowanie założonych efektów kształcenia realizowane jest przez:
  - monitorowanie i premiowanie pracy własnej studentów polegającej na samodzielnym rozwiązywaniu zadań algorytmicznych
  - premiowanie aktywności studentów na wykładach
- b) - kontrolę postępów studenta w rozwiązywaniu problemów algorytmicznych na zajęciach laboratoryjnych oraz poza nimi, które oceniane są przez automatyczny system sprawdzający

Ocena podsumowująca:

- a) w zakresie wykładów weryfikowanie założonych efektów kształcenia realizowane jest przez:
  - kolokwium wymagające rozwiązania 4-10 praktycznych problemów z zakresu algorytmiki,
  - zadania punktowane są w skali 0-5 punktów (maksymalna liczba punktów zależna od stopnia trudności), ze skokiem co 0,5 punktu,
  - punkty za kolokwia przeliczane są na procenty poprzez podzielenie przez maksymalną liczbę punktów do zdobycia i wliczane z wagą 0.3 do końcowej oceny przedmiotu
- b) w zakresie laboratoriów weryfikowanie założonych efektów kształcenia realizowane jest przez:
  - analizę wyników zadań rozwiązywanych w czasie semestru - ok. 30 zadań obowiązkowych, z czego na 3.0 trzeba rozwiązać 50% zadań, na 3.5 60% zadań, itd. - ocena wlicza się do końcowej oceny przedmiotu z wagą 0.7

Dodatkowe elementy oceny:

- za rozwiązanie dodatkowych zadań wybranych przez prowadzącego student może dostać do 1,5% dodatkowych punktów za zadanie

#### Treści programowe

Treści programowe zawarte w ramach tego przedmiotu oprócz przedstawienia treści algorytmicznych wymaganych od absolwenta studiów I stopnia zawierają obszerny zakres tematów, który ma przygotować studentów do startów w algorytmicznych konkursach w programowaniu zespołowym zgodnych z formułą ACM ICPC. Zakłada się, że przedmiot ten, jako przedmiot obieralny, wybrany



zostanie głównie przez studentów zainteresowanych tego typu zawodami. Dlatego też wiele poruszanych aspektów omawiane jest na poziomie średniozaawansowanym i nastawione na ich praktyczne wykorzystanie przy rozwiązywaniu problemów pojawiających się między innymi na zawodach algorytmicznych.

Wykłady rozpoczynają się od przedstawienia systemu pracy w ciągu semestru, który oparty będzie o automatyczny system sprawdzający, co dla wielu studentów będzie prawdopodobnie nowym sposobem nauki. Dlatego też przedstawiana jest dokładna prezentacja systemu oraz odpowiedzi na najczęściej zadawane pytania. Następnie prezentowana jest koncepcja programowania zespołowego w oparciu o formułę ACM ICPC wraz ze szczegółowymi zasadami oraz omówieniem kompetencji społecznych, które należy wypracować, aby zwiększyć szansę osiągnięcia w nich sukcesu. Kolejnym tematem są zagadnienia wprowadzające w algorytmikę poprzez omówienie podstawowych terminów z zakresu algorytmiki, takich jak problem i algorytm, dane i operacje na danych, instancja, pojęcie typu. Poruszona zostaje tematyka poprawności algorytmów, jej definiowanie oraz weryfikacja. Zagadnienia te obrazowane są prostymi algorytmami takimi jak wyszukiwanie binarne i jego zastosowanie do analizy ciągów posortowanych, ciąg sum i różnic częściowych, szybkie potęgowanie i wyznaczanie liczb Fibonacciego w czasie logarytmicznym, obliczanie wartości wielomianów, znajdowanie liczb pierwszych i największego wspólnego dzielnika. Na podstawie poznanych algorytmów omówiona zostaje złożoność obliczeniowa problemów oraz złożoność czasowa i pamięciowa algorytmów wraz ze sposobami jej wyznaczania, porównywania oraz zapisywania w notacji  $O()$ ,  $\Omega()$  i  $\Theta()$ . Poruszony zostaje problem złożoności w najgorszym i najlepszym przypadku oraz złożoność średnia. Następnie prezentowane są algorytmy sortowania począwszy od najprostszych, działających ze złożonością kwadratową, takich jak sortowanie bąbelkowe, przez wybór i wstawianie, przez szybsze sortowanie QuickSort, przez scalanie oraz Shella, po sortowania w czasie liniowym za pomocą algorytmu kubełkowego, pozycyjnego i przez zliczanie.

## Metody dydaktyczne

1. Wykład: prezentacja multimedialna ilustrowana przykładami podawanymi na tablicy, rozwiązywanie zadań.
2. Ćwiczenia laboratoryjne: rozwiązywanie zadań, ćwiczenia praktyczne, dyskusja, programowanie zespołowe.

## Literatura

### Podstawowa

1. Wprowadzenie do algorytmów, T.H. Cormen, Ch.E. Leiserson, R.L. Rivest, C. Stein, PWN, 2012
2. Algorytmika praktyczna nie tylko dla mistrzów, P. Stańczyk, PWN, 2009
3. Kombinatoryka dla programistów, W. Lipski, WNT, 2007
4. Tablice matematyczne, W. Mizerski, Adamantan, 2008
5. Symfonia C++ Standard, J. Grębosz, Edition 2000, 2008



Uzupełniająca

1. The CRC Concise Encyclopedia of Mathematics, E. W. Weisstein, CRC Press, 1998
2. Pasja C++, J. Grębosz, Edition 2000, 2010

**Bilans nakładu pracy przeciętnego studenta**

	Godzin	ECTS
Łączny nakład pracy	100	4,0
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	32	2,0
Praca własna studenta <sup>1</sup>	68	2,0

<sup>1</sup> niepotrzebne skreślić lub dopisać inne czynności