



## KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Zaawansowane programowanie [S2Bioinf1>ZPROG]

### Przedmiot

Kierunek studiów  
Bioinformatyka

Rok/Semestr  
1/1

Studia w zakresie (specjalność)

Profil studiów  
ogólnoakademicki

–

Poziom studiów  
drugiego stopnia

Język oferowanego przedmiotu  
polski

Forma studiów  
stacjonarne

Wymagalność  
obligatoryjny

### Liczba godzin

Wykład  
30

Laboratorium  
30

Inne (np. online)  
0

Ćwiczenia  
0

Projekty/seminaria  
0

### Liczba punktów ECTS

5,00

### Koordynatorzy

### Wykładowcy

### Wymagania wstępne

Studenci podejmujący studia na II stopniu Bioinformatyki powinni mieć osiągnięte efekty kształcenia z I stopnia tego kierunku studiów, zdefiniowane w Uchwale Senatu PP – efekty te prezentowane są w serwisie internetowym wydziału [cat.put.poznan.pl](http://cat.put.poznan.pl). W szczególności, studenci rozpoczynający ten przedmiot powinni posiadać wiedzę i umiejętności z zakresu następujących przedmiotów z I stopnia Bioinformatyki: Algorytmy i struktury danych, Podstawy programowania, Programowanie obiektowe.

### Cel przedmiotu

1. Przekazanie studentom podstawowej wiedzy o platformie programistycznej .NET. 2. Zapoznanie studentów z zasadami konstrukcji wielomianowych algorytmów metaheurystycznych oraz wykładniczych algorytmów dokładnych. 3. Wykształcenie u studentów umiejętności samodzielnego projektowania oraz implementowania zaawansowanych algorytmów dla złożonych problemów kombinatorycznych wywodzących się z biologii obliczeniowej, z wykorzystaniem platformy .NET.

### Przedmiotowe efekty uczenia się

Wiedza:

W wyniku przeprowadzonych zajęć student zna:

1. metody, techniki i narzędzia wykorzystywane w procesie rozwiązywania złożonych zadań bioinformatycznych, głównie o charakterze inżynierskim,

2. zasady tworzenia i testowania złożonego oprogramowania dla problemu bioinformatycznego,
3. zasady planowania badań z zakresu opracowywania nowych rozwiązań algorytmicznych dla problemów bioinformatycznych trudnych obliczeniowo.

Umiejętności:

W wyniku przeprowadzonych zajęć student umie:

1. planować i wykonywać zaawansowane eksperymenty obliczeniowe oraz interpretować ich wyniki,
2. stosować zaawansowane techniki i narzędzia informatyczne do rozwiązywania problemów biologicznych oraz ocenić ich przydatność,
3. dyskutować wyniki swoich prac w środowisku naukowym,
4. przygotować obszerne pisemne opracowanie w języku polskim przedstawiające wyniki własnych badań naukowych,
5. zastosować podejście systemowe do rozwiązania zadań bioinformatycznych, z uwzględnieniem aspektów pozatechnicznych.

Kompetencje społeczne:

Zaliczenie przedmiotu oznacza, że student jest gotów do:

1. określania priorytetów służących realizacji zadania zdefiniowanego przez siebie lub innych,
2. wzięcia odpowiedzialności za podejmowane decyzje.

### 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 odpowiedzi na pytania dotyczące materiału omówionego na poprzednich wykładach.
- b) W zakresie laboratoriów weryfikowanie założonych efektów kształcenia realizowane jest przez:
  - ocenę umiejętności związanych z realizacją ćwiczeń laboratoryjnych
  - ocenianie ciągle, na każdych zajęciach (odpowiedzi ustne) – premiowanie przyrostu umiejętności posługiwania się poznanymi zasadami i metodami
  - ocenę i „obronę” zrealizowanych przez studenta ćwiczeń laboratoryjnych

Ocena podsumowująca:

- a) w zakresie wykładów weryfikowanie założonych efektów kształcenia realizowane jest przez ocenę wiedzy i umiejętności związanych z treściami przekazywanymi na wykładach w formie jednego zbiorczego kolokwium.
- b) W zakresie laboratoriów weryfikowanie założonych efektów kształcenia realizowane jest przez ocenę wiedzy i umiejętności związanych z treściami przekazywanymi na wykładach i laboratoriach dokonywaną na podstawie przedstawionego przez studenta projektu końcowego oraz sprawozdania (uwzględnia się oceny wystawione w trakcie semestru).

### Treści programowe

Program wykładów obejmuje następujące zagadnienia.

Wykłady rozpoczynają się od odświeżenia wiedzy nt. elementów języka C#: środowiska programowego .NET: jego architektury, podstawowych cech .NET oraz innych związanych z językiem C# konstrukcji wymienionych w niniejszym paragrafie. Między innymi są to: struktura programu, rodzaje danych oraz operacje na nich, modularyzacja programu – definiowanie i wywoływanie funkcji, klasy, programowanie obiektowe w C#: hermetyzacja, dziedziczenie, polimorfizm, interfejsy, pliki i serializacja. Klasy kolekcji i udostępniane przez nie funkcje. Parametryczne klasy kolekcji (generics). Delegacje: definiowanie typu, tworzenie i wykorzystanie obiektu delegacji. Tablice delegacji, delegacje wielokrotne, delegacje anonimowe, wzorce delegacji. Wyrażenia lambda. Zdarzenia: definiowanie zdarzenia powiązanego z delegacją, rejestrowanie funkcji w zdarzeniu, zgłaszanie zdarzenia. Obsługa wyjątków. Współbieżność: tworzenie wątków współbieżnych i zarządzanie tymi wątkami. Współbieżne wykonywanie zadań, współbieżne wykonywanie obliczeń. Biblioteka Windows Forms. Wykorzystywane przestrzenie nazw, formularz początkowy i jego właściwości. Dodawanie elementów sterujących, określanie ich właściwości i definiowanie funkcji obsługi zdarzeń. Obsługa myszki i klawiatury, menu, pasek statusu, pasek narzędzi. Podstawowe elementy sterujące: przyciski, pola tekstowe, listy rozwijalne itp. Tworzenie wykresów. Okna dialogowe: dialogi standardowe i niestandardowe.

Seria wykładów prezentujących sposoby projektowania algorytmów rozpoczyna się przypomnieniem podstawowych pojęć z zakresu optymalizacji kombinatorycznej oraz wyjaśnieniem terminów heurystyka i metaheurystyka. Algorytmy genetyczne: ogólne zasady, parametry, reprezentacja osobnika, funkcja przystosowania, generowanie populacji początkowej, selekcja, krzyżowanie, mutacja, strategia intensyfikacji i dywersyfikacji. Metoda przeszukiwania tabu: ogólne zasady, parametry, generowanie rozwiązania początkowego, definiowanie sąsiedztwa, definiowanie ruchu, lista tabu, kryterium aspiracji, rodzaje pamięci, strategia intensyfikacji i dywersyfikacji. Heurystyki: konstrukcyjne vs. polepszające, heurystyka zachłanna, heurystyka przeszukiwania wiązkowego, heurystyka dekompozycji problemu, heurystyka lokalnego przeszukiwania, hiperheurystyka. Inne metaheurystyki: genealogia i klasyfikacja, GRASP, programowanie ewolucyjne, programowanie genetyczne, scatter search, ant colony optimization, particle swarm optimization, heurystyki hybrydowe. Algorytmy dokładne: ograniczenia stosowania, metoda podziału i ograniczeń (branch & bound), metoda podziału i odcięć (branch & cut), programowanie dynamiczne.

Ćwiczenia laboratoryjne prowadzone są w formie piętnastu dwugodzinnych zajęć odbywających się w laboratorium komputerowym. Pierwsze zajęcia przeznaczone są na zapoznanie studentów z zasadami użytkowania laboratorium i zaliczania ćwiczeń. Ćwiczenia realizowane są jednoosobowo. Program zajęć laboratoryjnych obejmuje następujące zagadnienia.

Przygotowanie aplikacji konsolowych, w których używane są podstawowe elementy języka C#. Opracowanie programów zmodularyzowanych wykorzystujących klasy kolekcji, interfejsy, delegacje. Opracowanie programów korzystających z plików dyskowych. Przygotowanie programów korzystających z biblioteki Windows Forms.

W ramach zajęć laboratoryjnych realizowany jest ponadto projekt końcowy. Realizacja projektu obejmuje zaprojektowanie, zaimplementowanie i przetestowanie jednej metaheurystyki – algorytmu genetycznego albo metody przeszukiwania tabu, o wielomianowej złożoności czasowej – dla jednego wybranego problemu bioinformatycznego spośród podanych. Metaheurystyki mają być specjalizowane, tzn. dostosowane do właściwości konkretnego problemu w celu optymalizacji jakości otrzymywanych rozwiązań. Dodatkowo projekt obejmuje implementację procedury generującej nietrywialne instancje w sposób losowy, z parametrami pozwalającymi na utworzenie instancji o różnym stopniu trudności, zwłaszcza dużych. Procedura ta umożliwi dogłębne przetestowanie możliwości i ograniczeń metaheurystyki, a wyniki obszernego eksperymentu obliczeniowego wchodzi w skład sprawozdania.

## Tematyka zajęć

brak

## Metody dydaktyczne

1. Wykład: prezentacja multimedialna, prezentacja ilustrowana przykładami podawanymi na tablicy.
2. Ćwiczenia laboratoryjne: przygotowywanie programów w środowisku .NET umożliwiających poznanie poszczególnych technologii występujących w tym środowisku; dyskusja metod realizacji projektów końcowych.

## Literatura

Podstawowa

1. Pro C# 5.0 and the .NET 4.5 Framework, 6th edition, Andrew Troelsen, apress, 2012
2. Metaheuristics: From Design to Implementation, El-Ghazali Talbi, Wiley, Hoboken, 2009
3. Jak to rozwiązać czyli nowoczesna heurystyka, Zbigniew Michalewicz, David B. Fogel, WNT, Warszawa, 2006

Uzupełniająca

1. Programowanie w C#, wydanie VI, Ian Griffiths, Matthew Adams, Jessy Liberty, O'Reilly 2010, Helion 2012
2. Algorytmy genetyczne i ich zastosowania, David E. Goldberg, WNT, Warszawa, 1995
3. Tabu Search, Fred Glover, Manuel Laguna, Kluwer Academic Publishers, Boston, 1997
4. Złożoność obliczeniowa problemów kombinatorycznych, Jacek Błażewicz, WNT, Warszawa, 1988

## Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	125	5,00
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	60	2,50
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych/ćwiczeń, przygotowanie do kolokwiiw/egzaminu, wykonanie projektu)	65	2,50