



KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Języki skryptowe w bioinformatyce

Przedmiot

Kierunek studiów

Bioinformatyka

Studia w zakresie (specjalność)

Poziom studiów

pierwszego stopnia

Forma studiów

stacjonarne

Rok/semestr

2/3

Profil studiów

ogólnoakademicki

Język oferowanego przedmiotu

polski

Wymagalność

obligatoryjny

Liczba godzin

Wykład

15

Laboratoria

30

Inne (np. online)

Ćwiczenia

Projekty/seminaria

Liczba punktów ECTS

4

Wykładowcy

Odpowiedzialny za przedmiot/wykładowca:

dr inż. Maciej Miłostan

Wydział Informatyki i Telekomunikacji

Odpowiedzialny za przedmiot/wykładowca:

Wymagania wstępne

Student rozpoczynający ten moduł powinien posiadać podstawową wiedzę o paradygmatach programowania strukturalnego i obiektowego, znać co najmniej jeden język programowania obiektowego i strukturalnego (preferowane C++/C). Powinien posiadać wiedzę z zakresu tworzenia struktur danych, w tym złożonych, podstawową wiedzę z zakresu algorytmiki i złożoności problemów kombinatorycznych. Powinien posiadać umiejętność rozwiązywania podstawowych problemów algorytmicznych i dokonywania analizy ich złożoności, powinien posiadać umiejętność wyszukiwania informacji.

Cel przedmiotu

1. Przekazanie studentom podstawowej wiedzy o systemach operacyjnych w zakresie umożliwiającym swobodne posługiwanie się systemem Linux i automatyzację zadań przy użyciu mechanizmów powłoki systemowej oraz języków skryptowych.
2. Przekazanie studentom wiedzy o języku Python i środowiskach deweloperskich w zakresie umożliwiającym implementację własnych klas i bibliotek.



3. Zapoznanie studentów z praktycznym zastosowaniem języków skryptowych do analiz danych biologicznych, w tym z wykorzystaniem bibliotek Biopython.
4. Rozwinięcie u studentów umiejętności rozwiązywania problemów wymagających przetwarzania danych biologicznych pochodzących z różnorodnych źródeł – umiejętność konwersji danych. W tym umiejętności przetwarzania sekwencji znaków i plików tekstowych z wykorzystaniem wyrażeń regularnych.
5. Wykształcenie u studentów umiejętności tworzenia własnych narzędzi w zgodzie z zadaną specyfikacją.

Przedmiotowe efekty uczenia się

Wiedza

W wyniku przeprowadzonych zajęć student:

1. Pogłębia wiedzę z zakresu programowania strukturalnego i obiektowego w kontekście języka Python.
2. Zdobywa podstawową wiedzę z zakresu praktycznego użytkowania systemów operacyjnych oraz wbudowanych w nie mechanizmów. Jest zaznajomiony z podstawowymi mechanizmami przetwarzania i automatyzacji danych dostępnymi w systemie Linux (przetwarzanie potokowe, podstawowe skrypty powłoki, przekierowanie wejścia/wyjścia).
3. Zna metody dostępu do baz danych (w tym zasobów publicznych) i plików z wykorzystaniem języka Python.
4. Posiada ugruntowaną i podbudowaną teorią wiedzę z zakresu wykorzystania wyrażeń regularnych i wzorców do przetwarzania danych tekstowych. Jest zaznajomiony z operacjami wyszukiwania ciągów znaków, modyfikacją ciągów, konwersją i formatowaniem tekstu, kodowaniem (ASCII, UTF8, BASE64), ekstrakcją danych zagregowanych/statystycznych. Potrafi korzystać z różnych źródeł danych tekstowych (pliki, bazy danych itp.).
5. Zna biblioteki bioinformatyczne i wie jak je wykorzystać do przetwarzania zarówno sekwencji biologicznych jak i struktur przestrzennych białek i kwasów nukleinowych. Wie jak z wykorzystaniem bibliotek Pythona pozyskiwać dane z publicznych baz danych (GenBank, PDB etc.).
6. Wie jak tworzyć własne biblioteki i jak ich używać.

Umiejętności

W wyniku przeprowadzonych zajęć student:

1. Stosuje podstawowe techniki i narzędzia informatyczne, dostępne w postaci bibliotek Pythona, do analizy danych biologicznych w postaci plików tekstowych z sekwencjami aminokwasowymi oraz danymi strukturalnymi (tekstowe pliki PDB). Potrafi pozyskiwać dane z baz danych biologicznych z poziomu programów w Pythonie. Potrafi dokonywać podstawowych przekształceń sekwencji



biologicznych i podstawowych operacji na plikach PDB i fasta. Umiejętności te są niezbędne do analizy złożonych problemów biologicznych.

2. Potrafi zastosować funkcje, struktury programistyczne, biblioteki oraz klasy dostępne w Pythonie do implementacji własnych narzędzi realizujących cele zgodne z zadaną specyfikacją.

Kompetencje społeczne

1. Potrafi odpowiednio określić priorytety służące realizacji zadania określonego przez siebie lub innych. Elementem niezbędnym do zaliczenia jest terminowa realizacja szeregu zadań praktycznych, w szczególności programistycznych.

2. Ma świadomość odpowiedzialności za podejmowane decyzje – braki w realizacji zadań, nieterminowe ich wykonanie lub próby plagiatu wpływają na uzyskiwane oceny.

Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

Wiedza nabyta w ramach wykładu jest weryfikowana za pomocą 60-minutowego kolokwium realizowanego na 15 wykładzie. Kolokwium składa się z 25-30 pytań (testowych i otwartych), różnie punktowanych. Próg zaliczeniowy: 50% punktów. Zagadnienia zaliczeniowe, na podstawie których opracowywane są pytania zostaną przesłane studentom drogą mailową z wykorzystaniem systemu uczelnianej poczty elektronicznej.

Umiejętności nabyte w ramach zajęć laboratoryjnych weryfikowane są na podstawie sprawozdań a także kolokwium przy komputerze, składającym się z 5 zadań różnie punktowanych w zależności od stopnia ich trudności oraz na podstawie implementacji projektu w języku Python realizującego wybrane analizy danych bioinformatycznych np. pochodzących z bazy PDB. Próg zaliczeniowy: 50% punktów.

Treści programowe

Program wykładu obejmuje następujące zagadnienia.

Wykład 1-2: Dane biologiczne a sekwencje i teksty – zarysowanie kontekstu i wskazanie obszarów, w którym stosuje się języki skryptowe. Podkreślenie znaczenia danych sekwencyjnych w biologii i wskazanie elementarnych narzędzi, które mogą zostać wykorzystane do ich analizy. Krótkie omówienie instrukcji Linuksa oraz wprowadzenie do języków skryptowych i interpreterów, pokazanie elementów programowania w powłoce i AWK.

Wykład 3-5: Paradygmaty programowania i dobre praktyki programistyczne w kontekście Pythona. Przygotowanie środowiska programistycznego. Omówienie składni języka, notacji i jego filozofii. Omówienie podstawowych typów danych i idei obiektowości w Pythonie. Wskazanie sposobów dokumentacji kodu i metod korzystania z niej. Ewolucja języka i podstawowe różnice w funkcjonalności między dwoma głównymi wersjami Pythona. Definiowanie własnych klas i mechanizm dziedziczenia. Funkcje anonimowe (lambda wyrażenia).



Wykład 6-8: Przykłady analiz danych z wykorzystaniem bibliotek bioinformatycznych w szczególności z wykorzystaniem Biopython. Implementacja interfejsu graficznego z wykorzystaniem wbudowanej w Pythona biblioteki Tkinter. Test zaliczeniowy.

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 przez studentów indywidualnie. Program zajęć laboratoryjnych obejmuje następujące zagadnienia. Wprowadzenie do pracy w systemie Linux – podstawowe instrukcje, przetwarzanie potokowe, przekierowywanie strumieni wejściowych i wyjściowych. Elementy programowania w powłoce – skrypty bash, elementy przetwarzania tekstów przy użyciu programów w AWK. Zapoznanie się z interpreterem Pythona oraz systemami pomocy i dokumentacji stanowiącymi część środowiska programistycznego. Implementacja programów mających na celu poznanie typów danych i struktur kontrolnych języka. Zapoznanie z mechanizmami wyrażeń regularnych i ich zastosowanie do analizy danych w plikach tekstowych z uwzględnieniem plików z sekwencjami i strukturami biologicznymi (pliki z bazy PDB). Przeprowadzenie testów zaimplementowanych programów na zbiorach danych testowych. Implementacja programów pozyskujących dane z publicznie dostępnych baz biologicznych i dokonujących ich przekształceń zgodnie z zadaną specyfikacją przy użyciu bibliotek Biopython. Projekt i implementacja własnych klas dokonujących operacji na danych biologicznych w szczególności sekwencjach i strukturach biologicznych.

Metody dydaktyczne

1. Wykład: prezentacja multimedialna wedle potrzeby ilustrowana dodatkowymi przykładami podawanymi na tablicy
2. Ćwiczenia laboratoryjne: ćwiczenia praktyczne przy komputerze realizowane według podanego scenariusza, implementacja programów i skryptów rozwiązujących zadane problemy, dyskusja zastosowanych rozwiązań i konstrukcji programistycznych

Literatura

Podstawowa

1. Bioinformatics Programming using Python / Mitchell L Model; O'Reilly 2010
2. Python. Wprowadzenie. Wydanie V (tytuł oryginału: Learning Python, 5th Edition) / Mark Lutz; Tł. Grzegorz Kowalczyk, Andrzej Watrak, Anna Trojan, Marek Pętlicki, Helion 2020.
3. Python 3 : kompletne wprowadzenie do programowania / Mark Summerfield ; [tł. Robert Górczyński]., Gliwice, Helion 2010.
4. Linux Pocket Guide, 2nd Edition, Daniel J. Barrett, O'Reilly Media 2012 (Dostępne tłumaczenie na język polski: Linux. Leksykon Kieszonkowy, Tł. Adam Bąk, Helion 2013)
5. The Linux Command Line: A Complete Introduction, William E. Shotts, Jr., No Starch Press, 2012 (dostępna również jako darmowy ebook: <http://linuxcommand.org/tlcl.php>)



6. System operacyjny LINUX- przewodnik użytkownika, Cezary Sobaniec, Nakom 2002
7. Linux in a Nutshell, 6th Edition, Ellen Siever, Stephen Figgins, Robert Love, Arnold Robbins, O'Reilly Media 2009

Uzupełniająca

1. Mastering Regular Expressions, Jeffrey E.F. Friedl, O'Reilly Media, 2006
2. Sed & Awk, Dougherty and Arnold Robbins, O'Reilly and Associates, 1997

Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	100	4,0
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	45	2,0
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych, przygotowanie do kolokwiów, wykonanie projektu) ¹	55	2,0

¹ niepotrzebne skreślić lub dopisać inne czynności