



ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Перший (освітній)</i>
Галузь знань	16 – Хімічна та біоінженерія
Спеціальність	162 Біотехнології та біоінженерія
Освітня програма	<i>Біотехнологія</i>
Статус дисципліни	Базова
Форма навчання	Денна
Рік підготовки, семестр	2 курс, осінній семестр
Обсяг дисципліни	Загальна кількість 120 год.
Семестровий контроль/ контрольні заходи	Залік/МКР
Розклад занять	<i>Лекції: 2 год/тиждень; Практичні: 2 год/тиждень</i>
Мова викладання	Українська
Інформація про керівника курсу / викладачів	Лектор: д. т. н., проф. Горобець Світлана Василівна, gorobetssv@gmail.com Практичні д. т. н., проф. Горобець Світлана Василівна, gorobetssv@gmail.com к. т. н., ас. Дем'яненко Ірина Володимирівна, iryna.demjanenko@gmail.com ас. Лахнеко Ольга Романівна, olhashnurenko@gmail.com
Розміщення курсу	

Програма навчальної дисципліни

1. Опис навчальної дисципліни, її мета, предмет вивчення та результати навчання

Сучасний світ вже не можливо уявити без інформаційних технологій, які увійшли у всі галузі. Професійний біотехнолог наразі має знати та вміти використовувати сучасні мови програмування. Лідером серед таких мов є Python, особливо Biopython, що є колекцією некомерційних інструментів для обчислення в біології та біоінформатиці з відкритим кодом створеною міжнародною асоціацією розробників Biopython - це набір модулів Python, які надають функції для роботи з послідовностями ДНК, РНК і білкових послідовностей, такими як зворотна комплементация ланцюжка ДНК, пошук мотивів в білкових послідовностях тощо. Він надає безліч синтаксичних аналізаторів для читання всіх основних генетичних баз даних, таких як GenBank, SwissPort, PDB, а також оболонки / інтерфейси для запуску інших популярних програм / інструментів біоінформатики, таких як NCBI BLASTN, Entrez тощо, в середовищі Python. Даний курс знайомить майбутніх біотехнологів з основами біоінформатики та програмування на мові Python.

Метою навчальної дисципліни є формування у студентів здатностей:

- використання інформаційних і комунікаційних технологій;
- використовувати знання з математики та фізики в обсязі, необхідному для досягнення інших результатів освітньої програми;
- використовувати сучасні автоматизовані системи управління виробництвом біотехнологічних продуктів різного призначення, їх технічне, алгоритмічне, інформаційне і програмне забезпечення для вирішення професійних завдань;
- вчитися і оволодівати сучасними знаннями; комплексно аналізувати біологічні та біотехнологічні процеси на молекулярному та клітинному рівнях; використовувати знання

про шляхи біосинтезу практично цінних метаболітів для вдосконалення біотехнологій їх одержання

Основні завдання кредитного модуля.

Згідно з вимогами програми навчальної дисципліни студенти після засвоєння кредитного модуля мають продемонструвати такі результати навчання:

знання :

- основи мови програмування Python та Biopython
- знати базові біоінформатичні алгоритми вирівнювання нуклеотидних та амінокислотних послідовностей;
- знати принципи роботи програмних пакетів Opencv-python, Matplotlib, Sklearn.

уміння:

- програмувати на базовому рівні на мові Python та Biopython;
- вміти реалізовувати основні біоінформатичні алгоритми вирівнювання на мові Python;
- вміти використовувати пакет Matplotlib Python для побудови графіків та діаграм, аналізу експериментальних даних;
- вміти використовувати пакет Opencv-python для аналізу зображень, отриманих методами оптичної мікроскопії, скануючої зондової мікроскопії, скануючої електронної мікроскопії, використання фільтрів для корекції зображень, виділення контурів біооб'єктів тощо.
- вміти використовувати пакет Sklearn Python для задач регресійного аналізу;
- вміти застосовувати сучасні методи біоінформатики для розв'язання практичних задач, пов'язаних з дослідженням і проектуванням біотехнологічних процесів.

Програмні результати навчання:

- Вміти застосовувати сучасні математичні методи для розв'язання практичних задач, пов'язаних з дослідженням і проектуванням біотехнологічних процесів. Використовувати знання фізики для аналізу біотехнологічних процесів.
- Базуючись на знаннях про закономірності механічних, гідромеханічних, тепло- та масообмінних процесів та основні конструкторські особливості, вміти обирати відповідне устаткування у процесі проектування виробництв біотехнологічних продуктів різного призначення для забезпечення їх максимальної ефективності.

2. Пререквізити та постреквізити дисципліни (місце в структурно-логічній схемі навчання за відповідною освітньою програмою)

Місце в структурно-логічній схемі навчання забезпечується дисциплінами, такими як загально технічна дисципліна «Вища математика», а також базовий рівень володіння англійською мовою не нижче A2. У структурно-логічній площині програми підготовки бакалаврів з біотехнології дисципліна базується на попередньо вивчених дисциплінах в школі. Вивчення дисципліни впливає на подальший шлях здобувачів вищої освіти. На базі дисципліни «Інформаційні технології» в подальшому вивчають наступні дисципліни «Методи аналізу в біотехнології» та на блок дисциплін зв'язаних з проектуванням біотехнологічних виробництв.

3. Зміст навчальної дисципліни

Тема 1. Інтерпретована, об'єктно-орієнтована мова програмування Python.

Лекція 1. Різні мови програмування та сфери їх застосування. Об'єктно-орієнтований і системний підходи. Порівняння об'єктно-орієнтованого та процедурного (традиційного) підходів. Особливості мови Python. Пакети Python для вирішення задач біотехнології та біоінформатики.

Лекція 2. Базові поняття мови Python: базовий синтаксис, типи даних, математичні операції, логічні оператори, умовні оператори, цикли.

Лекція 3. Складні структури даних мови Python. Робота зі списками, словниками, кортежами, множинами (наборами).

Лекція 4. Робота з функціями в мові Python. Поняття про класи та модулі.

Тема 2. Основи біоінформатики. Базові алгоритми біоінформатики на мові Python.

Лекція 5. Формування біоінформатики як самостійної науки. Предмет і задачі біоінформатики.

Лекція 6. Методи динамічного програмування. Алгоритм глобального вирівнювання двох послідовностей.

Лекція 7. Створення функцій в мові Python та приклад їх реалізація в алгоритмі глобального вирівнювання двох послідовностей.

Лекція 8. Робота з файлами в мові Python. Основи пакету Biopython для роботи з базами даних.

Лекція 9. Модифікації алгоритму вирівнювання генетичних послідовностей. Алгоритми локального та псевдоглобального вирівнювання двох нуклеотидних послідовностей.

Лекція 10. Модифікації алгоритму вирівнювання генетичних послідовностей. Алгоритми швидкого вирівнювання двох нуклеотидних послідовностей. Алгоритм побудови точкової матриці (dot-матриці).

Лекція 11. Система премій і штрафів для вирівнювання послідовностей. Алгоритм загальної функції штрафу.

Лекція 12. Хешування. Використання хеш-таблиці для пошуку співпадінь по списку слів.

Лекція 13. Особливості алгоритмів вирівнювання білкових послідовностей. Матриці амінокислотних замінів PAM і BLOSUM.

Тема 3. Використання мови програмування Python для аналізу експериментальних даних в біотехнології.

Лекція 14. Пакет Matplotlib для побудови графіків та діаграм.

Лекція 15. Пакет Sklearn для задач регресійного аналізу. Лінійна регресія експериментальних даних.

Лекція 16. Пакет Opensv-python для аналізу зображень, отриманих методами оптичної мікроскопії, скануючої зондової мікроскопії, скануючої електронної мікроскопії. Зчитування зображень біооб'єктів із файлів, їх запис у файл, використання фільтрів для корекції зображень, виділення контурів біооб'єктів тощо.

Лекція 17. Використання пакету Biopython для дослідження властивостей ДНК, РНК та білків на прикладі вірусу COVID-19.

Лекція 18. Біоінформатика як інструмент отримання нових знань.

Надається перелік розділів і тем всієї дисципліни.

Навчальні матеріали та ресурси

Базова література:

1. Горобець, С. В. Основи біоінформатики [Електронний ресурс]: підручник для студентів напряму підготовки 6.051401 «Промислова біотехнологія» факультету біотехнології і біотехніки / С. В. Горобець, О. Ю. Горобець, Т. А. Хоменко ; НТУУ «КПІ». - Електронні текстові дані (1 файл: 2,72 Мбайт). – Київ : НТУУ «КПІ», 2010
2. Горобець С. В. Біоінформатика. Практикум [Електронний ресурс]: навч. посіб. для студ. спеціальності 162 «Біотехнології та біоінженерія» / С. В. Горобець, О. Ю. Горобець, І.В. Дем'яненко; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл 5.49 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2020. – 86с. <https://ela.kpi.ua/handle/123456789/38813>
3. Горобець С. В. Біоінформатичні бази даних [Електронний ресурс] : навч. посіб. для студ. спеціальності 162 «Біотехнології та біоінженерія» / С. В. Горобець, О. Ю. Горобець, М. О. Булаєвська ; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 3,86 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2020. – 117 с. <https://ela.kpi.ua/handle/123456789/36457>

4. Замуруєва О. В., Кримусь А. С., Ольхова Н. В. Об'єктно-орієнтоване програмування в Python : курс лекцій. Луцьк : Вежа-Друк, 2018. – 64 с.
5. Яковенко А.В. Основи програмування. Python. Частина 1 [Електронний ресурс]: підручник для студ. спеціальності 122 "Комп'ютерні науки", спеціалізації "Інформаційні технології в біології та медицині" / А. В. Яковенко ; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 1,59 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2018. – 195 с.

Додаткова:

1. Пекарський Б.Г. «Основи програмування. Навчальний посібник». – Кондор, 2018. – С. 368.
2. Основи програмування на мові Python : комплексний навчальний посібник : в 2 томах / І.В. Мельник. – Київ : Кафедра, 2020. – 2 томи.
3. Мова програмування Python для інженерів і науковців : навчальний посібник / В.Б. Копей; Міністерство освіти і науки України, Івано-Франківський національний технічний університет нафти і газу, кафедра комп'ютеризованого машинобудування. – Івано-Франківськ : ІФНТУНГ, 2019. – 267 сторінок.

Інформаційні ресурси

1. <https://www.ncbi.nlm.nih.gov/>
2. <https://www.python.org/>
3. <https://matplotlib.org/>
4. <https://scikit-learn.org/stable/>
5. https://docs.opencv.org/4.5.2/d6/d00/tutorial_py_root.html

Навчальний контент

4. Методика опанування навчальної дисципліни (освітнього компонента)

1	2
	<i>Тема 1. Інтерпретована, інтегрована, об'єктно-орієнтована мова програмування Python.</i>
1	Лекція 1. Різні мови програмування та сфери їх застосування. Об'єктно-орієнтований і системний підходи. Порівняння об'єктно-орієнтованого та процедурного (традиційного) підходів. Література: базова [4,5], додаткова [1-3], інформаційні джерела [2]
2	Лекція 2. Особливості мови Python. Пакети Python для вирішення задач біотехнології та біоінформатики. Інтегральне середовище розробки (ICP), (<i>Integrated development environment – IDE</i>) Pycharm Community. Література: базова [4,5], додаткова [1-3], інформаційні джерела [2]
3	Лекція 3. Типи даних в Python. Математичні операції, умовні оператори, цикли. Зчитування та запис файлів. Література: базова [4,5], додаткова [1-3], інформаційні джерела [2]
4	Лекція 4. Робота з функціями в мові Python. Поняття про класи та модулі. Література: базова [4,5], додаткова [1-3], інформаційні джерела [2]
	<i>Тема 2. Основи біоінформатики. Базові алгоритми біоінформатики на мові Python.</i>
5	Лекція 5. Формування біоінформатики як самостійної науки. Предмет і задачі біоінформатики. Література: базова [1-3]
6	Лекція 6. Методи динамічного програмування. Алгоритм глобального вирівнювання двох послідовностей. Література: базова [1-3]

7	Лекція 7. Використання пакету Biopython для роботи з базами даних. Основні бази даних біотехнологічної інформації. Література: базова [4,5], додаткова [1-3], інформаційні джерела [2]
8	Лекція 8. Створення функцій, класів та проєктів у Python приклад їх реалізація в алгоритмі глобального вирівнювання двох послідовностей. Література: базова [4,5], додаткова [1-3], інформаційні джерела [2]
9	Лекція 9. Модифікації алгоритму вирівнювання генетичних послідовностей. Алгоритми локального та псевдоглобального вирівнювання двох нуклеотидних послідовностей. Література: базова [1-3]
10	Лекція 10. Модифікації алгоритму вирівнювання генетичних послідовностей. Алгоритми швидкого вирівнювання двох нуклеотидних послідовностей. Алгоритм побудови точкової матриці (dot-матриці). Література: базова [1-3].
11	Лекція 11. Система премій і штрафів для вирівнювання послідовностей. Алгоритм загальної функції штрафу. Література: базова [1-3].
12	Лекція 12. Хешування. Використання хеш-таблиці для пошуку співпадінь по списку слів. Література: базова [1-3].
13	Лекція 13. Особливості алгоритмів вирівнювання білкових послідовностей. Матриці амінокислотних замінів PAM і BLOSUM. Література: базова [1- 3]
	<i>Тема 3. Використання мови програмування Python для аналізу експериментальних даних в біотехнології.</i>
14	Лекція 14. Пакет Matplotlib для побудови графіків та діаграм. Література: базова [3-5], інформаційні джерела [2-4]
15	Лекція 15. Пакет Sklearn для задач регресійного аналізу. Лінійна регресія експериментальних даних. Література: базова [3-5], інформаційні джерела [2-4]
16	Лекція 16. Пакет Opensv-python для аналізу зображень, отриманих методами оптичної мікроскопії, скануючої зондової мікроскопії, скануючої електронної мікроскопії. Зчитування зображень біооб'єктів із файлів, їх запис у файл, використання фільтрів для корекції зображень, виділення контурів біооб'єктів тощо. Література: базова [4,5], додаткова [1-3], інформаційні джерела [2,5]
17	Лекція 17. Використання пакету Biopython для дослідження властивостей ДНК, РНК та білків на прикладі вірусу COVID-19. Література: базова [4,5], додаткова [1-3], інформаційні джерела [2]
18	Лекція 18. Біоінформатика як інструмент отримання нових знань. Література базова: [1-3].

Практичні заняття

Основні завдання циклу практичних занять:

- розробка елементарних програм в Python
- робота з програмними пакетами, які використовуються в біотехнології;
- практична робота з сучасними базами даних молекулярної біології.

1.	Практична робота 1 Знайомство та основи роботи в базах даних Національного центру біотехнологічної інформації США (National Center for Biotechnology Information (NCBI)). Література: базова [1-3], інформаційні ресурси [1].
----	--

2.	Практична робота 2. Загальна характеристика основних біологічних молекул (ДНК, РНК та білків), як об'єктів біоінформатичних досліджень. Література: базова [1-3], інформаційні ресурси [1].
3.	Практична робота 3. Налаштування середовища розробки та запуск програми на Python. Математичні операції. Введення та виведення даних. Література: базова [4,5], додаткова [1-3] інформаційні джерела [2]
4.	Практична робота 4. Умовні оператори, цикли. Зчитування та запис файлів. Література: базова [4,5], додаткова [1-3] інформаційні джерела [2]
5.	Практична робота 5. Робота з функціями в мові Python. Література: базова [4,5], додаткова [1-3] інформаційні джерела [2]
6.	Практична робота 6 Алгоритм глобального вирівнювання та його реалізація на мові Python. Програмування прямого шляху розрахунку Література: базова [1-5], додаткова [1-3] інформаційні джерела [2]
7.	Практична робота 7 Алгоритм глобального вирівнювання та його реалізація на мові Python. Програмування зворотного шляху алгоритм Література: базова [1-5], додаткова [1-3] інформаційні джерела [2]
8.	Практична робота 8 Алгоритм глобального вирівнювання та його реалізація на мові Python. Перевірка та пошук багів в програмі. Література: базова [1-5], додаткова [1-3] інформаційні джерела [2]
9.	Практична робота 9 Алгоритм локального вирівнювання та його реалізація на мові Python. Модифікація алгоритму глобального вирівнювання. Література: базова [1-5], додаткова [1-3] інформаційні джерела [2]
10.	Практична робота 10 Алгоритм локального вирівнювання та його реалізація на мові Python. Перевірка та пошук багів у програмі. Література: базова [1-5], додаткова [1-3] інформаційні джерела [2]
11.	Практична робота 11. Порівняння послідовностей з використанням програми BLAST. Програми родини BLAST. Література: базова [4-5], інформаційні ресурси [1].
12.	Практична робота 12. Пошук гомологів різних білків в BLAST та порівняння результатів з іншими програмами. Література: базова [4-5], інформаційні ресурси [1].
13.	Практична робота 13 Алгоритм хешування. Література: базова [1-3]
14.	Практична робота 14 Робота з пакетами Matplotlib для побудови графіків та діаграм. Література: базова [4-5], інформаційні ресурси [1].

15.	Практична робота 15 Пакет Sklearn для задач регресійного аналізу. Лінійна регресія експериментальних даних. Література: базова [4-5], інформаційні ресурси [1].
16.	Практична робота 16 Пакет OpenCV-python для аналізу зображень, отриманих методами оптичної мікроскопії, скануючої зондової мікроскопії, скануючої електронної мікроскопії. Зчитування зображень біооб'єктів із файлів, їх запис у файл, використання фільтрів для корекції зображень, виділення контурів біооб'єктів тощо. Література: Література: базова [1-4], додаткова [1-3] інформаційні джерела [2-5]
17.	Практична робота 17 МКР
18.	Практична робота 18 Залік

Самостійна робота студента

Самостійна робота студента по дисципліні включає підготовку до аудиторних занять (48 годин), модульної контрольної (4 години), написання ДКР (10 годин), підготовка до заліку (6 годин).

Політика та контроль

5. Політика навчальної дисципліни (освітнього компонента)

Вивчення дисципліни «Інформаційні технології» відбувається на лекційних та практичних заняттях. Наочність навчальних занять забезпечується використанням значної кількості ілюстративного матеріалу (схем, таблиць, слайдів). Під час викладання даної дисципліни викладач проводить опитування здобувачів для того, щоб визначити рівень засвоєння ними викладеного матеріалу, важливим є активність здобувачів. Практичні заняття проходять з використанням комп'ютерної техніки та відповідного програмного забезпечення.

Положення про рейтингову систему оцінювання з дисципліни «Інформаційні технології» до даної робочої навчальної програми представлені в пункті 9 силабуса.

Зазначається система вимог, які викладач ставить перед студентом/аспірантом:

- *правила відвідування занять (як лекцій, так і практичних/лабораторних);*
Відвідування лекцій, практичних занять та лабораторних робіт, а також відсутність на них, не оцінюється. Однак, студентам рекомендується відвідувати заняття, оскільки на них викладається теоретичний матеріал та розвиваються навички, необхідні для формування компетентностей, визначених стандартом освіти. Система оцінювання орієнтована на отримання балів за активність студента, а також виконання завдань, які здатні розвинути практичні уміння та навички. За об'єктивних причин (наприклад, хвороба, працевлаштування, міжнародне стажування тощо) навчання може відбуватися в он-лайн формі за погодженням із керівником курсу.
- *правила поведінки на заняттях (активність, підготовка коротких доповідей чи текстів, відключення телефонів, використання засобів зв'язку для пошуку інформації на гугл-диску викладача чи в інтернеті тощо);*
На аудиторних заняттях студент має поважати викладача та дисципліну, що він слухає; Виконувати елементарні правила та норми поведінки; Протягом заняття забороняється користуватися мобільними телефонами, окрім екстрених випадків. Норми етичної поведінки студентів і працівників визначені у розділі 2 Кодексу честі Національного

технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського». Детальніше: <https://kpi.ua/code>.

- *правила призначення заохочувальних та штрафних балів;*
за активну роботу на лекціях студент може отримати до 10 балів сумарно протягом семестру.
- *політика дедлайнів та перескладань;*
Термін здачі кожного виду роботи обговорюється на занятті під час видачі завдання та залежить від типу роботи. Роботи, які здаються із порушенням термінів без поважних причин, оцінюються на нижчу оцінку. Перескладання тем (модулів) відбувається за наявності поважних причин.
- *політика щодо академічної доброчесності;*
визначені у розділі 3 Кодексу честі Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського». Детальніше: <https://kpi.ua/code>. Використання додаткових джерел інформації під час оцінювання знань заборонено (у т.ч. мобільних девайсів). Мобільні пристрої дозволяється використовувати лише під час он-лайн тестування та виконання розрахунків.
інші вимоги, що не суперечать законодавству України та нормативним документам Університету.

6. Види контролю та рейтингова система оцінювання результатів навчання (PCO)

Поточний контроль: виконання практичних робіт (55 балів), виконання домашньої контрольної роботи (15 балів) написання експрес-тестів (20 балів) та МКР (10 балів). Загальна сума балів за семестрову роботу – 100 балів. Докладніша інформація щодо поточного контролю та критеріїв оцінювання наведена в PCO з дисципліни. (Додаток 1).

Календарний контроль: два рази за семестр

Семестровий контроль: залік. Загальна сума балів на заліку – 100 балів. Докладніша інформація щодо проведення та оцінювання наведена в PCO з дисципліни.

Умови допуску до семестрового контролю: семестровий рейтинг не нижче 50 балів, написання МКР, захист усіх практичних робіт, написання експрес-тестів та задача ДКР. Якщо семестровий рейтинг є вищим за 60 балів за умови здачі всіх видів робіт студент може отримати залік автоматом.

7. Додаткова інформація з дисципліни (освітнього компонента)

Додаток 1

Система рейтингових (вагових) балів занять і рейтингових оцінок по видах контролю за рік

№ п/п	Вид контролю	Бал	Кількість	Сума балів
1	Виконання практичних робіт			
	- ваговий бал Γ_k^*	5	11	55
	-якість виконання*	0-5		
2.	Експрес-контроль	4	5	20
	- ваговий бал Γ_k^*	0-4		
	-якість виконання**			
3.	Модульна контрольна робота			
	-ваговий бал Γ_k	10	1	10
	- якість виконання***	0-10		
4.	ДКР			
	-ваговий бал Γ_k	15	1	15
	- якість виконання***	0-15		
5.				100

* - Якість виконання практичних робіт:

бездоганна робота	– 5 балів;
є певні недоліки у підготовці та/або виконанні роботи	– 4 бали;
є суттєві недоліки у підготовці та/або виконанні роботи	– 3 бали;
Робота не виконана або не захищена	– 0-2 балів.

** - Якість виконання експрес-контролю:

повна розкрита відповідь	– 3,5-4 балів ;
помилка в одному завданні або неповна відповідь в двох завданнях	– 2,5-3 балів ;
помилка в двох завдань або неповна відповідь в 4 завданнях	– 1,5-2 балів;
робота не зарахована	– 0 -1 балів.

*** - Якість виконання модульної контрольної роботи. :

повна розкрита відповідь	– 9-10 балів ;
помилка в одному завданні або неповна відповідь в двох завданнях	– 8 балів ;
помилка в двох завдань або неповна відповідь в 4 завданнях	– 7 балів;
робота не зарахована	– 0-6 балів.

*** - Якість виконання домашньої контрольної роботи. :

повна розкрита відповідь	– 14-15 балів ;
помилка в одному завданні або неповна відповідь в двох завданнях	– 12-13 балів ;
помилка в двох завдань	– 10-11 балів;
робота не зарахована	– 0 -9 балів.

Розрахунок шкали (R) рейтингу

Сума вагових балів контрольних заходів протягом семестру складає:

$$R = 55 + 20 + 10 + 15 = 100 \text{ балів};$$

Рейтингова шкала з дисципліни складає $R = 100$ балів;

Необхідною умовою для одержання заліку автоматом є зарахування усіх пропозицій, що виносяться на обговорення виконання на позитивну оцінку модульної контрольної роботи та загальний рейтинг більше 60 балів. Для підвищення оцінки проводиться залікова робота. Попередній рейтинг анулюється.

Календарний контроль:

Рубіжні (планові атестації). Студент повинен набрати балів:: 1 атестація – «зараховано» - 20 балів (40 – максимум), 2 атестація – 40 балів (80 – максимум).

Підсумкова оцінка якості знань з дисципліни визначаються за традиційною 6-рівневою шкалою на базі індивідуальних поточних оцінок за такою шкалою:

Рейтинг	Оцінка ESTS	Традиційна оцінка
$95 \leq R < 100$	A	Відмінно
$85 \leq R < 95$	B	Дуже добре
$75 \leq R < 85$	C	Добре
$65 \leq R < 75$	D	Задовільно
$60 \leq R < 65$	E	Достатньо
$R < 60$	Fx	Незадовільно

Семестровий контроль: залік. Загальна сума балів заліку – 100 балів. Умови допуску до семестрового контролю: семестровий рейтинг не менше 50 балів, написання МКР та ДКР, виконання практичних робіт.

Заліковий білет складається з 5 питань, 1 питання оцінюється у 20 балів.

Повна відповідь на питання – (20) балів

Зроблені незначні помилки – (17-19) балів

Суттєві помилки у відповіді – (12-16) балів

Відповіді не вірні – (0-11) бали.

Таблиця відповідності рейтингових балів оцінкам за університетською шкалою:

<i>Кількість балів</i>	<i>Оцінка</i>
100-95	Відмінно
94-85	Дуже добре
84-75	Добре
74-65	Задовільно
64-60	Достатньо
Менше 60	Незадовільно

Додаток 2

Орієнтовний перелік тем ДКР

1. Вбудовані функції Python
2. Вбудовані константи Python
3. Вбудовані типи даних Python
4. Вбудовані виключення Python
5. Модулі Python
6. Модулі для обробки тексту Python
7. Модулі для роботи з різними типами даних Python
8. Числові та математичні модулі Python
9. Модулі для функціонального програмування Python
10. Модулі доступу до файлів і директорій Python
11. Збереження даних Python
12. Стиснення та архівація даних Python
13. Формати файлів Python
14. Модулі для взаємодії з операційною системою Python
15. Сервіси періоду виконання Python
16. Модулі для програмних фреймворків Python
17. Модулі для створення графічного інтерфейсу користувача з Tk Python
18. Криптографічні сервіси Python
19. Модулі для паралельного програмування (н-д, threading, multiprocessing, subprocess, sched, queue та ін.) Python
20. Модулі для мережевого та міжпроцесового спілкування Python
21. Модулі для обробки даних Інтернету Python
22. Модулі обробки структурованої розмітки Python
23. Модулі для підтримки Інтернет-протоколів Python
24. Модулі для мультимедійних послуг Python
25. Модулі для інтернаціоналізації Python
26. Модулі для засобів розробки Python
27. Модулі для відладки та профілювання Python
28. Модулі для публікації та встановлення програмного забезпечення Python

Додаток 3

Питання до контрольної роботи з курсу «Інформаційні технології»

1. Історія розвитку та основні мови програмування.

2. Об'єктно-орієнтований підхід до програмування.
3. Процедурний підхід до програмування.
4. Основні переваги Python. Задачі, для яких застосовують Python.
5. Типи даних в Python.
6. Логічні операції в Python.
7. Математичні оператори та функції в Python.
8. Функції в Python.
9. Класи в Python.
10. Основні пакети Python.
11. Основні функції роботи зі списками.
12. Основні функції роботи зі словниками.
13. Основні функції роботи з кортежами.
14. Основні функції роботи з множинами (наборами).
15. Визначення поняття біоінформатики.
16. Предмет біоінформатики.
17. Цілі біоінформатики.
18. Мета програми «Геном людини».
19. Коли стартувала програми «Геном людини» та основні її досягнення?
20. Коли використовується алгоритм локального вирівнювання для порівняння двох послідовностей?
21. Коли використовується алгоритм глобального вирівнювання для порівняння двох послідовностей?
22. Коли використовується алгоритм псевдоглобального вирівнювання для порівняння двох послідовностей?
23. Коли використовується алгоритм швидкого вирівнювання для порівняння двох послідовностей?
24. Що означає кожна комірка матриці ваг префіксів $a(i,j)$?
25. Основна ідея алгоритмів динамічного програмування?
26. Бази даних (загальне визначення) і біоінформаційні БД. В яких областях молекулярної біології застосовують БІБД.
27. Історія виникнення БІБД. Найбільш відомі організації - куратори БІБД і створені ними банки даних.
28. Класифікація БІБД. Наведіть приклади.
29. Які методи використовуються для вирівнювання 2-х послідовностей, коли і ким ці методи були розроблені?
30. Коли використовується алгоритм загальної функції штрафу для порівняння двох послідовностей?
31. Алгоритм загальної функції штрафу.
32. Ініціалізація масиву ваг префіксів $a(i,j)$; $b(i,j)$; $c(i,j)$ при використанні загальної функції штрафу.
33. Рекурентна формула для глобального вирівнювання двох послідовностей.
34. Рекурентна формула для локального вирівнювання двох послідовностей.
35. Рекурентна формула для псевдоглобального вирівнювання двох послідовностей.
36. Рекурентна формула для загальної функції штрафу.
37. Ініціалізувати нульовий стовпчик та нульову строку для глобального вирівнювання двох послідовностей.
38. Ініціалізувати нульовий стовпчик та нульову строку для локального вирівнювання двох послідовностей.
39. Ініціалізувати нульовий стовпчик та нульову строку для псевдоглобального вирівнювання двох послідовностей.
40. Ініціалізувати нульовий стовпчик та нульову строку для загальної функції штрафу.
41. Алгоритм хешування, в яких програмах він використовується?
42. Сформувати хеш-таблицю для дослідження послідовності (довжина слова 4).
43. Основні функції пакету Matplotlib для побудови графіків.
44. Основні функції пакету Opencv-python для роботи з зображеннями біологічних об'єктів графіків.

45. Основні засоби роботи з генетичними послідовностями в пакету Biopython.
46. Функції для лінійної регресії в пакеті Sklearn.

Додаток 4

Питання до заліку з курсу «Інформаційні технології»

1. Історія розвитку та основні мови програмування.
2. Об'єктно-орієнтований підхід до програмування.
3. Процедурний підхід до програмування.
4. Основні переваги Python. Задачі, для яких застосовують Python.
5. Типи даних в Python.
6. Логічні операції в Python.
7. Математичні оператори та функції в Python.
8. Функції в Python.
9. Класи в Python.
10. Основні пакети Python.
11. Основні функції роботи зі списками.
12. Основні функції роботи зі словниками.
13. Основні функції роботи з кортежами.
14. Основні функції роботи з множинами (наборами).
15. Визначення поняття біоінформатики.
16. Предмет біоінформатики.
17. Цілі біоінформатики.
18. Мета програми «Геном людини».
19. Коли стартувала програми «Геном людини» та основні її досягнення?
20. Коли використовується алгоритм локального вирівнювання для порівняння двох послідовностей?
21. Коли використовується алгоритм глобального вирівнювання для порівняння двох послідовностей?
22. Коли використовується алгоритм псевдоглобального вирівнювання для порівняння двох послідовностей?
23. Коли використовується алгоритм швидкого вирівнювання для порівняння двох послідовностей?
24. Що означає кожна комірка матриці ваг префіксів $a(i,j)$?
25. Основна ідея алгоритмів динамічного програмування?
26. Бази даних (загальне визначення) і біоінформаційні БД. В яких областях молекулярної біології застосовують БІБД.
27. Історія виникнення БІБД. Найбільш відомі організації - куратори БІБД і створені ними банки даних.
28. Класифікація БІБД. Наведіть приклади.
29. Які методи використовуються для вирівнювання 2-х послідовностей, коли і ким ці методи були розроблені?
30. Коли використовується алгоритм загальної функції штрафу для порівняння двох послідовностей?
31. Алгоритм загальної функції штрафу.
32. Ініціалізація масиву ваг префіксів $a(i,j)$; $b(i,j)$; $c(i,j)$ при використанні загальної функції штрафу.
33. Рекурентна формула для глобального вирівнювання двох послідовностей.
34. Рекурентна формула для локального вирівнювання двох послідовностей.
35. Рекурентна формула для псевдоглобального вирівнювання двох послідовностей.
36. Рекурентна формула для загальної функції штрафу.
37. Ініціалізувати нульовий стовпчик та нульову строку для глобального вирівнювання двох послідовностей.
38. Ініціалізувати нульовий стовпчик та нульову строку для локального вирівнювання двох послідовностей.

39. Ініціалізувати нульовий стовпчик та нульову строку для псевдоглобального вирівнювання двох послідовностей.
40. Ініціалізувати нульовий стовпчик та нульову строку для загальної функції штрафу.
41. Алгоритм хешування, в яких програмах він використовується?
42. Сформувати хеш-таблицю для дослідження послідовності (довжина слова 4).
43. Основні функції пакету Matplotlib для побудови графіків.
44. Основні функції пакету Opencv-python для роботи з зображеннями біологічних об'єктів графіків.
45. Основні засоби роботи з генетичними послідовностями в пакету Biopython.
46. Функції для лінійної регресії в пакеті Sklearn.