



## ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Робоча програма навчальної дисципліни (Силабус)

### Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Перший (бакалаврський)</i>
Галузь знань	16 – Хімічна та біоінженерія
Спеціальність	162 Біотехнології та біоінженерія
Освітня програма	<i>Біотехнології</i>
Статус дисципліни	Базова
Форма навчання	Денна
Рік підготовки, семестр	2 курс, осінній семестр
Обсяг дисципліни	Загальна кількість 4 кредити / 120 год.
Семестровий контроль/ контрольні заходи	Залік/ МКР/ ДКР
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, <a href="mailto:gorobetssv@gmail.com">gorobetssv@gmail.com</a> , телеграм-чат «Інформаційні технології: очна форма навчання» <a href="https://t.me/+c8Y6Zqg4sts2MDgy">https://t.me/+c8Y6Zqg4sts2MDgy</a> Практичні д. т. н., проф. Горобець Світлана Василівна, <a href="mailto:gorobetssv@gmail.com">gorobetssv@gmail.com</a> , телеграм-чат «Інформаційні технології: очна форма навчання» <a href="https://t.me/+c8Y6Zqg4sts2MDgy">https://t.me/+c8Y6Zqg4sts2MDgy</a>
Розміщення курсу	На платформі дистанційного навчання Google Клас за посиланням <a href="https://classroom.google.com/c/NDA4MTY0OTYzOTA3?cjc=3ku5mbm">https://classroom.google.com/c/NDA4MTY0OTYzOTA3?cjc=3ku5mbm</a> код курсу 3ku5mbm, постійне посилання на Google Meet <a href="https://meet.google.com/uom-mkaz-tem">https://meet.google.com/uom-mkaz-tem</a>

### 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 Matplotlib, Opencv-python, 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 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 з рядками, списками, словниками, кортежами, множинами (наборами).

Лекція 4. Робота з файлами в мові Python.

Лекція 5. Основи пакету numpy мови програмування Python.

Лекція 6. Функції в Python.

Лекція 7. Класи та проєкти в Python.

## **Тема 2. Використання мови програмування Python для аналізу експериментальних даних в біотехнології.**

Лекція 8. Пакет Matplotlib для побудови графіків та діаграм.

Лекція 9. Пакети Python для задач регресійного аналізу. Лінійна регресія експериментальних даних.

Лекція 10. 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.

## **Тема 3. Основи біоінформатики. Базові алгоритми біоінформатики на мові Python.**

Лекція 11. Формування біоінформатики як самостійної науки. Предмет і задачі біоінформатики.

Лекція 12. Основи пакету Biopython для роботи з базами даних. Попарне та множинне вирівнювання послідовностей в Biopython. Організація доступу до бази даних NCBI Entrez в пакеті Biopython.

Лекція 13. Методи динамічного програмування. Алгоритм глобального вирівнювання двох послідовностей та приклад реалізації алгоритму глобального вирівнювання двох послідовностей на мові програмування Python.

Лекція 14. Модифікації алгоритму вирівнювання генетичних послідовностей. Алгоритми локального, псевдоглобального та швидкого вирівнювання двох послідовностей.

Лекція 15. Хешування. Використання хеш-таблиць в базах даних для пошуку співпадінь по списку слів. Алгоритм побудови точкової матриці (dot-матриці).

Лекція 16. Особливості алгоритмів вирівнювання білкових послідовностей. Матриці амінокислотних замін PAM і BLOSUM, PSSM матриці.

Лекція 17. Система премій і штрафів для вирівнювання послідовностей. Алгоритм загальної функції штрафу.

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.  
<https://ela.kpi.ua/bitstream/123456789/774/1/10-11-146.pdf>
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. 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.

Додаткова:

1. Основи програмування на мові Python : комплексний навчальний посібник : в 2 томах / І.В. Мельник. – Київ : Кафедра, 2020. – 2 томи.
2. 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.

### 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](https://docs.opencv.org/4.5.2/d6/d00/tutorial_py_root.html)
6. <http://www.biopython.org>

### Навчальний контент

#### 4. Методика опанування навчальної дисципліни (освітнього компонента)

1	2
	<i>Тема 1. Інтерпретована, інтегрована, об'єктно-орієнтована мова програмування Python.</i>
1	<b>Лекція 1.</b> 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. Література: базова [4,5], додаткова [1-2], інформаційні джерела [1-6]
2	<b>Лекція 2.</b> Базові поняття мови Python: базовий синтаксис, типи даних, математичні операції, логічні оператори, умовні оператори, цикли. Література: базова [4,5], додаткова [1-2], інформаційні джерела [2]
3	<b>Лекція 3.</b> Складні структури даних мови Python. Робота з рядками, списками, словниками, кортежами, множинами (наборами). Література: базова [4,5], додаткова [1-2], інформаційні джерела [2]
4	<b>Лекція 4.</b> Робота з файлами в мові Python. Література: базова [4,5], додаткова [1-1], інформаційні джерела [2]
5.	<b>Лекція 5.</b> Основи пакету pipru мови програмування Python. Література: базова [4,5], додаткова [1-2], інформаційні джерела [2]

6.	<b>Лекція 6.</b> Функції в Python. Види функцій, аргументи функцій, документація функцій тощо. Література: базова [4,5], додаткова [1-2], інформаційні джерела [2]
7.	<b>Лекція 7.</b> Класи та проекти в Python. Література: базова [4,5], додаткова [1-2], інформаційні джерела [2]
	<i>Тема 2. Використання мови програмування Python для аналізу експериментальних даних в біотехнології.</i>
14	<b>Лекція 8.</b> Пакет Matplotlib для побудови графіків та діаграм. Література: базова [5], інформаційні джерела [3]
15	<b>Лекція 9.</b> Пакети Python для задач регресійного аналізу. Лінійна регресія експериментальних даних. Література: базова [5], інформаційні джерела [2-5]
16	<b>Лекція 10.</b> 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 тощо. Література: базова [5], інформаційні джерела [2,5]
	<i>Тема 3. Основи біоінформатики. Базові алгоритми біоінформатики на мові Python.</i>
8	<b>Лекція 11.</b> Формування біоінформатики як самостійної науки. Предмет і задачі біоінформатики. Література: базова [1-3].
9	<b>Лекція 12.</b> Основи пакету Biopython для роботи з базами даних. Попарне та множинне вирівнювання послідовностей в Biopython. Організація доступу до бази даних NCBI Entrez в пакеті Biopython. Література: базова [1-3], інформаційні джерела [1, 6]
10	<b>Лекція 13.</b> Методи динамічного програмування. Алгоритм глобального вирівнювання двох послідовностей та приклад реалізації алгоритму глобального вирівнювання двох послідовностей на мові програмування Python. Література: базова [1-3], інформаційні джерела [1]
11	<b>Лекція 14.</b> Модифікації алгоритму вирівнювання генетичних послідовностей. Алгоритми локального, псевдоглобального та швидкого вирівнювання двох послідовностей. Література: базова [1-3], інформаційні джерела [1]
12	<b>Лекція 15.</b> Хешування. Використання хеш-таблиць в базах даних для пошуку співпадінь по списку слів. Алгоритм побудови точкової матриці (dot-матриці). Література: базова [1-3], інформаційні джерела [1]
13	<b>Лекція 16.</b> Особливості алгоритмів вирівнювання білкових послідовностей. Матриці амінокислотних замін PAM і BLOSUM, PSSM матриці. Література: базова [1- 3], інформаційні джерела [1]
17	<b>Лекція 17.</b> Система премій і штрафів для вирівнювання послідовностей. Алгоритм загальної функції штрафу. Література: базова [1-3], інформаційні джерела [1]
18	<b>Лекція 18.</b> Біоінформатика як інструмент отримання нових знань. Література базова: [1-3], інформаційні джерела [1]

### 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.	<p><b>Практична робота 1</b></p> <p>Знайомство та основи роботи в базах даних Національного центру біотехнологічної інформації США (National Center for Biotechnology Information (NCBI)). Робота з базами даних PMC та PubMed.</p> <p>Література: базова [1-3], інформаційні ресурси [1].</p>
2.	<p><b>Практична робота 2.</b></p> <p>Налаштування середовища розробки та запуск програми на Python. Введення та виведення даних. Математичні операції. Умовні оператори, цикли.</p> <p>Література: базова [4,5], додаткова [1,2] інформаційні джерела [2]</p>
3.	<p><b>Практична робота 3.</b></p> <p>Робота з типами даних мови Python: рядками, списками, словниками, кортежами, множинами (наборами).</p> <p>Література: базова [4,5], додаткова [1,2] інформаційні джерела [2]</p>
4.	<p><b>Практична робота 4.</b></p> <p>Робота з файлами в мові Python. Зчитування послідовностей з баз даних NCBI та запис їх у файли.</p> <p>Література: базова [4,5], додаткова [1,2] інформаційні джерела [2]</p>
5.	<p><b>Практична робота 5</b></p> <p>Основи пакету numpy мови програмування Python. Функції пакету numpy, робота з масивами.</p> <p>Література: базова [4,5], додаткова [1,2] інформаційні джерела [2]</p>
6.	<p><b>Практична робота 6</b></p> <p>Функції в Python, види функцій, локальні та глобальні змінні.</p> <p>Література: базова [4,5], додаткова [1,2] інформаційні джерела [2]</p>
7.	<p><b>Практична робота 7</b></p> <p>Класи та проєкти в Python.</p> <p>Література: базова [4,5], додаткова [1,2] інформаційні джерела [2]</p>
8.	<p><b>Практична робота 8</b></p> <p>Робота з пакетами Matplotlib для побудови графіків та діаграм.</p> <p>Література: базова [5], інформаційні ресурси [3].</p>
9.	<p><b>Практична робота 9</b></p> <p>Пакети мови програмування Python для задач регресійного аналізу. Лінійна регресія експериментальних даних.</p> <p>Література: базова [5], інформаційні ресурси [2-5].</p>
10.	<p><b>Практична робота 10</b></p> <p>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.</p> <p>Література: базова [5], інформаційні джерела [2-5]</p>
11.	<p><b>Практична робота 11.</b></p> <p>Порівняння послідовностей з використанням програми BLAST. Програми родини BLAST.</p> <p>Література: базова [1-3], інформаційні ресурси [1].</p>
12.	<p><b>Практична робота 12.</b></p> <p>Основи пакету Biopython для роботи з базами даних. Організація доступу до бази даних NCBI Entrez в пакеті Biopython.</p>

	Література: базова [5], інформаційні джерела [6]
13.	<b>Практична робота 13</b> Алгоритм глобального вирівнювання та його реалізація на мові Python. Література: базова [1-5], інформаційні джерела [2]
14.	<b>Практична робота 14</b> Алгоритм локального, псевдоглобального та швидкого вирівнювання двох послідовностей та реалізація алгоритму локального вирівнювання на мові Python. Література: базова [1-5], додаткова [1,2] інформаційні джерела [2]
15.	<b>Практична робота 15</b> Алгоритм хешування. Побудова хеш-таблиць. Література: базова [1-3], інформаційні ресурси [1].
16.	<b>Практична робота 16.</b> Особливості алгоритмів вирівнювання білкових послідовностей. Розрахунок матриці амінокислотних замінів PAM та матриці PSSM. Література: базова [1- 3], інформаційні джерела [2]
17.	<b>Практична робота 17</b> <b>МКР</b>
18.	<b>Практична робота 18</b> <b>Залік</b>

### 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 оцінюються. 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.*
- Студент зобов'язаний зареєструватися на платформі дистанційного навчання Google Workspace for Education Fundamentals (в минулому G Suit For Education) на домені @LLL.kpi.ua та приєднатися до Google Класу «Інформаційні технології» за посиланням <https://classroom.google.com/c/NDA4MTY0OTYzOTA3?cjc=3ku5mbm> . Для цього студенту необхідно спочатку отримати акаунт в Google Workspace for Education Fundamentals на домені @LLL.kpi.ua. Для отримання акаунту в Google Workspace for Education Fundamentals на домені @LLL.kpi.ua необхідно заповнити Google Форму: [https://sikorsky-distance.kpi.ua/reg\\_gsuite/](https://sikorsky-distance.kpi.ua/reg_gsuite/). Після реєстрації та модерації заявки студента, адміністратор надішле студенту на пошту пароль та логін до акаунту, з яким студент зможе використовувати всі доступні інструменти та сервіси Google Workspace for Education Fundamentals. Google Workspace for Education Fundamentals – це пакет спеціалізованого хмарного програмного забезпечення, інструментів для спільної роботи та дистанційного навчання від компанії Google. Основна складова пакету – система управління навчанням Google Клас, яка дозволяє викладачу створювати навчальні класи, оцінювати завдання, надавати учням зворотній зв'язок, публікувати оголошення і поширювати навчальні матеріали. Викладач може бачити, хто виконав завдання, а хто ще продовжує над ним працювати, а також читати питання і коментарі учнів. Для приєднання до навчального курсу «Інформаційні технології» студенту потрібно перейти у Google Клас за посиланням <https://classroom.google.com>, натиснути зображення «+» у верхньому правому кутку браузера, вибрати «Приєднатися до класу» та ввести код курсу 3ku5mbm. Акаунти студентів, які приєдналися до Google Класу не з акаунту на домені @LLL.kpi.ua, будуть вилучатися з навчального курсу «Інформаційні технології» Google Класу тому, що автоматичний імпорт оцінок за тестування можливий виключно з акаунту Інформаційні технології на домені @LLL.kpi.ua. Система Google Клас автоматично надсилає кожному студенту бали по кожному з видів контролю на електронну пошту. Тому для ознайомлення з балами за кожен окремий вид контролю студенту необхідно змінити налаштування електронної пошти так, щоб ці електронні листи не потрапляли у спам. Всі виконані завдання для перевірки викладачем студент повинен завантажувати через систему Google Клас (результати виконання завдань, надіслані через телеграм канал перевірятися не будуть);
- Листування із студентами з організаційних питань буде здійснюватися через телеграм-чат «Інформаційні технології: очна форма навчання» <https://t.me/+c8Y6Zqg4sts2MDgy>.



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

**Поточний контроль:** тестування знань студентів з лекційного матеріалу та практичних робіт (81 бал), виконання домашньої контрольної роботи (10 балів) та модульної контрольної роботи (9 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 експрес-тесту по МКР, написання експрес-тесту ДКР та всіх експрес-тестів по практичним роботам. Якщо семестровий рейтинг є вищим за 60 балів за умови здачі всіх видів робіт студент може отримати залік автоматом.

Перевірка виконання СРС здійснюватиметься оцінюванням результатів тестування на платформі дистанційного навчання Google Workspace for Education Fundamentals на домені @LLL.kpi.ua в системі Google Клас за посиланням:

<https://classroom.google.com/c/NDA4MTY0OTYzOTA3?cjc=3ku5mbm>.

Кожен з тестів можливо пройти тільки один раз. Рейтингові (вагові) бали занять і рейтингові оцінки по всіх видах контролю в Google Класі дорівнюють відповідним балам в силабусі з коефіцієнтом 10 для зручності розрахунку балів (щоб не використовувати дробові числа). Відповідно перед кожною атестацією, а також в кінці семестру всі набрані студентом бали в Google Класі будуть ділитися на 10 і вноситись до системи АІС «Електронний кампус» КПІ імені Ігоря Сікорського.

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
2.	Тестування знань студентів з лекційного матеріалу та практичних робіт	4,5	18	81
	-якість виконання*	0-4,5		
3.	Модульна контрольна робота			
	-ваговий бал $r_k$	9	1	9
	- якість виконання**	0-9		
4.	ДКР			
	-ваговий бал $r_k$	10	1	10
	- якість виконання***	0-10		
5.				100

\* - Якість виконання тестування з лекційного матеріалу та практичних робіт:

бездоганна робота – 4,5 балів;  
є певні недоліки у виконанні роботи – 3,4 бали;  
є суттєві недоліки у виконанні роботи – 2,8 бали;  
робота не виконана – 0-2,7 балів.

\*\* - Якість виконання модульної контрольної роботи:

бездоганна робота – 9,0-8,6 балів;  
є певні недоліки у виконанні роботи – 6,8-8,5 балів;

є суттєві недоліки у виконанні роботи  
робота не виконана

– 5,4-6,7 балів;  
– 0 – 5,3 балів.

\*\*\* - Якість виконання домашньої контрольної роботи:

бездоганна робота

– 9-10 балів ;

є певні недоліки у виконанні роботи

– 7 - 8 балів ;

є суттєві недоліки у виконанні роботи

– 6 балів;

робота не виконана

– 0-5 балів.

### Розрахунок шкали (R) рейтингу

Сума вагових балів контрольних заходів протягом семестру складає:

$$R = 81+10+9 = 100 \text{ балів:}$$

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. При цьому 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:

<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

**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.

Залік складається з 10 питань, 1 питання оцінюється у 10 балів.

бездоганна робота

– 9 - 10 балів

є певні недоліки у виконанні роботи

– 7-8 балів

є суттєві недоліки у виконанні роботи

– 2-6 балів

Відповіді не вірні

– 0-1 бал.

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

**Орієнтовний перелік питань для ДКР з курсу «Інформаційні технології»**

1. Вбудовані функції Python
2. Вбудовані типи даних Python
3. Пакет Python numpy
4. Пакет Python matplotlib
5. Робота з масивами пакету numpy
6. Аналіз експериментальних даних з застосуванням пакету numpy:  
середнє значення величини;  
середню квадратичну похибку середнього;  
середнє квадратичне відхилення величини;  
дисперсію величини;  
максимальне та мінімальне значення величини тощо.
7. Створення функцій в Python.
8. Побудова діаграм на основі експериментальних даних.
9. Оформлення осей діаграми.
10. Використання словника при оформленні діаграми.
11. Використання пакету numpy для підготовки даних для візуалізації даних.

**Перелік питань до модульної 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 Orncsv-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

#### Перелік питань 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 БД.
27. Історія виникнення БД. Найбільш відомі організації - куратори БД і створені ними банки даних.
28. Класифікація БД. 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.

Робочу програму навчальної дисципліни (силабус):

складено д.т.н., проф. Горобець С.В.

Ухвалено кафедрою біоенергетики, біоінформатики та екобіотехнології (протокол № 15 від 29.06.2022 р.)

Погоджено Методичною комісією факультету (протокол № 9 від 30.06.22).