

**БІЛГОРОД-ДНІСТРОВСЬКИЙ ФАХОВИЙ КОЛЕДЖ
ПРИРОДОКОРИСТУВАННЯ, БУДІВНИЦТВА ТА
КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ**

Циклова комісія інформаційних технологій



ЗАТВЕРДЖЕНО

**Заступник директора з
навчальної роботи**

Марина ЗАЙЧЕНКО

«29» серпня 2024 р.

ОСНОВИ ПРОГРАМУВАННЯ

**ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
здобувачів освіти спеціальності
121 «Інженерія програмного забезпечення»**

Білгород-Дністровський, 2024

Програма навчальної дисципліни «**Основи програмування**» складена відповідно до освітньо-професійної програми для здобувачів освіти зі спеціальності **121 «Інженерія програмного забезпечення»**

Розробник: Боцан Марія Володимирівна, викладач інформаційних технологій, кваліфікаційна категорія «спеціаліст I категорії».

Робоча програма розглянута та схвалена на засіданні циклової комісії інформаційних технологій

Протокол № 1 від 29.08.2024 року

Голова циклової комісії  Сергій ТІТЯПКИН

Схвалено методичною радою Білгород-Дністровського фахового коледжу природокористування, будівництва та комп'ютерних технологій

Протокол № 6 від 29.08.2024 року

Голова методичної ради  Марина ЗАЙЧЕНКО

1. ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Найменування показників	Компонент освітньої програми, спеціальність, освітньо-професійний ступінь	Характеристика навчальної дисципліни	
		денна форма навчання	заочна форма навчання
Кількість кредитів: 7,5 Модулів: 2 Змістових модулів: 2 Загальна кількість годин: 255	Компонент освітньої програми <i>Цикл професійної підготовки</i> Спеціальність <i>121 «Інженерія програмного забезпечення»</i> Освітньо-професійний ступінь <i>«Фаховий молодший бакалавр»</i>	Обов'язкова	
		Рік підготовки:	
		2	-
		Семестр	
		3,4	-
		Лекційні заняття:	
		60 год.	-
		Лабораторні заняття	
		63 год.	-
		Самостійна робота	
		102 год.	-
		Курсова робота	
		30 год	-
		Індивідуальні заняття:	
		-	-
		Вид контролю:	
диференційований залік (3й семестр) / екзамен (4й семестр) (денна форма)			
-			

2. МЕТА ТА ЗАВДАННЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Навчальна дисципліна передбачена структурно-логічною схемою підготовки фахівців освітньо-професійного ступеню «Фаховий молодший бакалавр» за освітньо-професійною програмою «Інженерія програмного забезпечення».

Метою викладання навчальної дисципліни "Основи програмування" полягає у формуванні базових знань і навичок програмування, що дозволять студентам розуміти принципи побудови алгоритмів та розробки програмного забезпечення. Студенти навчатимуться вирішувати прикладні задачі шляхом створення програм, застосовуючи сучасні мови програмування, а також аналізувати і оптимізувати алгоритми для підвищення їх ефективності.

Завдання вивчення навчальної дисципліни "Основи програмування" полягає у формуванні базових знань з програмування та навичок створення алгоритмів для вирішення різноманітних завдань. Студенти повинні ознайомитися з основними поняттями програмування, такими як змінні, оператори, цикли та функції, основ ООП, а також навчитися використовувати ці елементи для написання простих програм. Важливим завданням є розвиток аналітичного мислення для ефективного вирішення проблем, пошуку та виправлення помилок у коді, а також оптимізації програм для покращення їх продуктивності.

Предметом вивчення дисципліни «Основи програмування» є ознайомлення студентів із базовими принципами та методами розробки програмного забезпечення. Вивчаються фундаментальні концепції програмування, такі як змінні, типи даних, оператори, умови, цикли, функції, масиви та алгоритми. Основною метою є формування вміння вирішувати задачі різної складності шляхом написання програм на основі алгоритмічного підходу, а також розвиток логічного та критичного мислення, що допоможе студентам застосовувати отримані знання на практиці.

Міждисциплінарні зв'язки: «Інформатика», «Математика», «Вступ до спеціальності», «Об'єктно-орієнтоване програмування», «Організація баз даних».

Набуті здобувачами освіти компетенції згідно з вимогами освітньо-професійної програми «Інженерія програмного забезпечення»:

ЗК05. Знання та розуміння предметної області та розуміння професійної діяльності.

ЗК07. Здатність застосовувати знання у практичних ситуаціях.

СК01. Здатність алгоритмічно та логічно мислити.

СК03. Здатність застосовувати теоретичні та емпіричні знання для розроблення, тестування, впровадження та супроводу програмного забезпечення.

СК04. Здатність спілкуватися іноземною мовою.

СК07. Здатність застосовувати знання у практичних ситуаціях.

СК09. Здатність вибирати та використовувати ефективні інструментальні засоби розробки програмного продукту.

Структура навчальної дисципліни є орієнтовною. Під час складання

навчальних програм викладачі навчальних закладів можуть вносити обґрунтовані зміни та доповнення в зміст програмного матеріалу і розподіл навчальних годин за темами в межах бюджетного часу, відведеному навчальним планом на вивчення дисципліни. Внесені зміни повинні бути обговорені на засіданні циклової комісії і затверджені заступником директора з навчальної роботи.

3. ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Модуль 1

ОСНОВИ ПРОГРАМУВАННЯ МОВОЮ PYTHON

Тема 1.1. Вступ до дисципліни

Ознайомлення з дисципліною. Ознайомлює студентів із базовими поняттями програмування на одній із найпопулярніших мов – Python.

Тема 1.2. Огляд засобів розробки ПЗ

Ознайомлення із основними інструментами, що використовуються для створення, тестування та налагодження програм. У цьому розділі розглядаються середовища розробки (IDE), текстові редактори, системи контролю версій (наприклад, Git), компілятори, інтерпретатори та відлагоджувачі.

Лабораторне заняття

Встановлення Python та середовища розробки. Проведення їх налаштування та тестування.

Тема 1.3. Засоби введення та виведення даних. Робота зі змінними та функцією print() в мові програмування Python

Ознайомлення з основними аспектами взаємодії програми з користувачем та обробки інформації. Вивчення того, як у Python можна оголошувати змінні, присвоювати їм значення різних типів (числа, рядки, булеві значення тощо), а також використовувати функцію print() для виведення результатів на екран.

Лабораторне заняття

Написання першого консольного додатку з використанням методів зчитування та виведення на Python.

Тема 1.4. Операції над числами в мові програмування Python.

Ознайомлення із основними математичними діями, які можна виконувати з числовими типами даних, такими як цілі числа (int) і числа з плаваючою комою (float).

Лабораторне заняття

Написання консольного додатку з використанням операцій над числами в Python.

Тема 1.5. Бібліотека математичних функцій Math

Ознайомлення із вбудованими функціями для виконання різноманітних математичних операцій.

Лабораторне заняття

Написання консольного додатку використанням математичних функцій.

Тема 1.6. Розгалуження, умовні оператори if elif else

Ознайомлення з основними механізмами прийняття рішень на основі умов. Вивчення умовних операторів if, elif, else.

Лабораторне заняття

Написання консольного додатку з реалізацією розгалужень на умовних операторах.

Тема 1.7. Принципи роботи з циклами. Основні відомості про цикли For та While

Ознайомлення з основними концепціями і принципами використання циклів для автоматизації повторюваних завдань. Цикли For та While.

Лабораторне заняття

Написання консольного додатку з використанням циклів For та While.

Тема 1.8. Робота з рядками в Python. Методи та функції рядків. Функція Range()

Ознайомлення з основами маніпуляцій з текстовими даними, а також використання функції range() для створення послідовностей чисел.

Лабораторне заняття

Написання консольного додатку з використанням методів та функцій рядків, а також з використанням функції Range().

Тема 1.9. Масиви. Списки та кортежі в Python

Ознайомлення з основними типами даних, які дозволяють зберігати та управляти колекціями значень. Вивчення списків та кортежів в Python.

Лабораторне заняття

Написання консольного додатку із застосуванням списків та кортежів, їх методів.

Тема 1.10. Основи роботи з двовимірними масивами в Python

Ознайомлення з основами організації та маніпуляції даними у формі матриць або таблиць. Використання списків для створення двовимірних масивів.

Лабораторне заняття

Написання консольного додатку для роботи з двовимірними масивами за допомогою списків.

Тема 1.11. Словники. Базові принципи роботи із словниками , порівняння із списками та кортежами

Ознайомлення з основами використання словників у Python, які є важливим типом даних для зберігання пар "ключ-значення".

Лабораторне заняття

Написання консольного додатку з використанням словників, кортежів та їх методів.

Тема 1.12. Множини в Python. Базові поняття та принципи роботи.

Вивчення основ роботи з множинами, які є важливим типом даних у Python для зберігання унікальних значень.

Лабораторне заняття

Написання консольного додатку з використанням множин та їх класів.

Тема 1.13. F-рядки в мові програмування Python. Переваги та недоліки

Синтаксис форматування рядків за допомогою F-рядків в мові програмування Python. Ознайомлення з перевагами та недоліками.

Лабораторне заняття

Написання консольного додатку з проведенням форматування за допомогою F-рядків.

Модуль 2

КОМПЛЕКСНЕ ПРОГРАМУВАННЯ МОВОЮ PYTHON

Тема 2.1. Робота с файлами в Python. Загальні поняття. Відкриття та закриття файлу.

Вивчення основ маніпуляції з файлами у Python, включаючи їх створення, читання, запис та зберігання даних у різних форматах.

Лабораторне заняття

Написання консольного додатку з використанням взаємодії з файлами та системою на Python.

Тема 2.2. Функції в Python. Базові принципи роботи з функціями. Їх призначення та принципи використання.

Вивчення основи функціонального програмування в Python. Створення та використання функції для спрощення коду та підвищення його читабельності.

Лабораторне заняття

Написання консольного додатку з реалізацією власних функцій із застосуванням параметрів та аргументів.

Тема 2.3. Вкладені функції. Значення параметрів за замовчуванням. Глобальні та локальні змінні. Рекурсії

Ознайомлення з вкладеними функціями, значеннями параметрів за замовчуванням, глобальними та локальними змінними, а також передачею значень у вкладені функції. Вивчення принципів створення рекурсії.

Лабораторне заняття

Написання консольного додатку з використанням вкладених функцій, параметрів по замовчуванню, локальних та глобальних змінних.

Тема 2.4. Lambda функції. Їх призначення та способи використання. Реалізація складної логіки у лямбда-виразах

Ознайомлення з концепцією анонімних функцій у Python, які дозволяють створювати прості функції без необхідності їх формального оголошення.

Лабораторне заняття

Написання консольного додатку з реалізацією Lambda функцій.

Тема 2.5. Базові поняття ООП. Поняття класу. Аргументи (параметри) класу. Об'єкти класу

Ознайомлення з основами ООП у Python, включаючи структуру класів і об'єктів, а також важливість параметрів класу.

Лабораторне заняття

Написання консольного додатку зі створенням перших простих класів та їх об'єктів.

Тема 2.6. Методи класу. Передача параметрів в клас.

Ознайомлення з методами, що визначають поведінку класів, і способи

передачі параметрів для налаштування об'єктів.

Лабораторне заняття

Написання консольного додатку з розширенням можливостей класів через використання методів та передачу параметрів в класи.

Тема 2.7. Поняття конструктору класу. Принцип роботи та взаємодії з ним. Магічні методи в Python.

Вивчення того, як конструктори класів використовуються для ініціалізації об'єктів, розгляд магічних методів, які надають спеціальну функціональність класам.

Лабораторне заняття

Написання консольного додатку з використанням конструкторів класів та магічних методів.

Тема 2.8. Спадкування в ООП. Інкапсуляція в ООП.

Ознайомлення з двома ключовими принципами об'єктно-орієнтованого програмування (ООП) – спадкуванням та інкапсуляцією, які допомагають структурувати та організовувати код.

Лабораторне заняття

Написання консольного додатку з використанням наслідування та інкапсуляції в класах.

Тема 2.9. Введення в Tkinter. Створення вікон.

Ознайомлення з модулем Tkinter та його встановленням. Вивчення принципів створення вікон у Tkinter.

Лабораторне заняття

Встановлення та налаштування бібліотеки Tkinter. Перевірка роботи через створення першого вікна додатку.

Тема 2.10. Введення в віджети. Tk и ttk. Обробка подій.

Ознайомлення з віджетами у Tkinter, Tk и ttk та обробкою подій у вікнах.

Лабораторне заняття

Написання віконного додатку з використанням віджетів та обробок подій.

Тема 2.11. Основні елементи вікон в Tkinter.

Ознайомлення з основними елементами вікон у Tkinter, такими як кнопки, написи, поля введення, прапори, радіокнопки, а також принципами їх

використання.

Лабораторне заняття

Написання віконного додатку з використанням основних елементів вікон в бібліотеці Tkinter.

Тема 2.12. Позиціонування елементів вікон в Tkinter.

Ознайомлення з позиціонуванням елементів у вікні за допомогою Pack, Place та Grid.

Лабораторне заняття

Написання віконного додатку з використанням позиціонування елементів у вікні.

Тема 2.13. Віджет Text в Tkinter.

Ознайомлення з віджетом Text в Tkinter та його принципом роботи та використання для створення текстових полів.

Лабораторне заняття

Написання віконного додатку з використанням віджета Text в Tkinter.

Тема 2.14. Віджет Treeview. Створення таблиць та дерев в Tkinter.

Ознайомлення з віджетом Treeview в Tkinter, його використанням для створення таблиць та ієрархічних дерев.

Лабораторне заняття

Написання віконного додатку з використанням таблиці та структурних дерев, створених за допомогою віджета Treeview.

Тема 2.15. Вікна. Стилізація вікон.

Ознайомлення з додатковими можливостями вікон, її стилізації, а також створення діалогових вікон та вікон-повідомлень.

Лабораторне заняття

Написання віконного додатку з використанням стилізації та додаткових вікон.

Тема 2.16. Підсумкове заняття. Аналіз пройденого матеріалу

Підведення підсумків з пройденого матеріалу, закріплення та перевірка за допомогою тестового контролю знань.

КУРСОВА РОБОТА

Мета виконання курсової роботи - закріплення та поглиблення теоретичних знань, набуття студентами навичок з розробки програмного забезпечення.

Тематику курсової роботи визначає циклова комісія навчального закладу згідно з рекомендаціями методичних об'єднань.

Виходячи з цього, завдання курсової роботи полягає в тому, щоб на основі отриманих під час вивчення знань провести проектування та реалізація віконного додатку на мові програмування Python, що забезпечує зручний інтерфейс для користувачів та реалізує визначені функціональні можливості. У рамках роботи передбачається дослідити сучасну бібліотеку для створення графічних інтерфейсів, таку як Tkinter, а також реалізувати принципи об'єктно-орієнтованого програмування. вміти провести самостійне створення віконного додатку та оформлення пояснювальну записку до курсової роботи.

Склад і оформлення курсової роботи

Курсова робота складається з розробленої програми і пояснювальної записки згідно до завдання курсової роботи та стандартів оформлення документації.

Програма

Програма має бути реалізована у повному об'ємі та виконувати завдання згідно до обраної теми курсової роботи. Відхилення допускаються лише в плані доповнення програми додатковим функціоналом, що має відповідати загальній темі додатку.

Програма обов'язково має:

- Бути виконана у вигляді віконного додатку
- Мати не менше двох функціональних вікон
- Забезпечувати обробку даних користувача
- Зберігати результат своєї роботи

Текстова частина

Розрахункова-пояснювальна записка оформляється в такому порядку і складається з розділів:

- зміст;
- загальні положення (вступ);
- постановку задачі для розробки;
- аналіз вимог до програми;
- побудову алгоритму програми;
- розробку програми;
- контрольне тестування програми;
- висновок;
- література.

Оформлення записки виконується згідно з вимогами єдиного графічного режиму навчального закладу. Всі таблиці і ілюстрації відповідно пронумерувати.

4. СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

«Основи програмування»

Назви розділів і тем	Кількість годин									
	Денна форма					Заочна форма				
	з а г а л ь н и й о б с я г	аудиторні			с а м о с т і й н а р о б о т а	з а г а л ь н и й о б с я г	аудиторні			с а м о с т і й н а р о б о т а
		в с ь о г о	з них				в с ь о г о	з них		
т е о р е т и ч н і			л а б о р а т о р н і	т е о р е т и ч н і				п р а к т и ч н і		
1	2	3	4	5	6	7	8	9	10	11
Модуль 1 Основи програмування мовою python										
Тема 1.1 Вступ до дисципліни.	5	1	1	0	4					
Тема 1.2 Огляд засобів розробки ПЗ.	8	4	2	2	4					
Тема 1.3 Засоби введення та виведення даних. Робота зі змінними та функцією print() в мові програмування Python.	8	4	2	2	4					
Тема 1.4 Операції над числами в мові програмування Python.	8	4	2	2	4					
Тема 1.5 Бібліотека математичних функцій Math.	8	4	2	2	4					
Тема 1.6 Розгалуження, умовні оператори if elif else.	8	4	2	2	4					
Тема 1.7 Принципи роботи з циклами. Основні відомості про цикли For та While.	8	4	2	2	4					
Тема 1.8 Робота з рядками в Python. Методи та функції рядків. Функція Range().	8	4	2	2	4					
Тема 1.9 Масиви. Списки та кортежі в Python.	8	4	2	2	4					
Тема 1.10 Основи роботи з двовимірними масивами в Python.	8	4	2	2	4					
Тема 1.11 Словники. Базові принципи роботи із словниками, порівняння із списками та кортежами.	10	6	2	4	4					

Тема 1.12 Множини в Python. Базові поняття та принципи роботи. Класи set та frozenset.	8	4	2	2	4					
Тема 1.13 F-рядки в мові програмування Python. Переваги та недоліки.	8	4	2	2	4					
Разом за змістовим модулем 1	103	51	25	26	52					
РАЗОМ за I семестр	103	51	25	26	52					
1	2	3	4	5	6	7	8	9	10	11
Модуль 2 Комплексне програмування мовою python										
Тема 2.1 Робота с файлами в Python. Загальні поняття. Відкриття та закриття файлу. Функції Open() і Close(). Модулі struct та pickle.	6	4	2	2	2					
Тема 2.2 Функції в Python. Базові принципи роботи з функціями. Їх призначення та принципи використання. використання параметрів та аргументів.	8	4	2	2	4					
Тема 2.3 Вкладені функції. Значення параметрів за замовчуванням. Глобальні та локальні змінні. Передача значень у вкладені функції. Рекурсії	8	4	2	2	4					
Тема 2.4 Lambda функції. Їх призначення та способи використання. Реалізація складної логіки у лямбда-виразах.	6	4	2	2	2					
Тема 2.5 Базові поняття ООП. Поняття класу. Аргументи (параметри) класу. Об'єкти класу.	8	6	4	2	2					
Тема 2.6 Методи класу. Передача параметрів в клас.	8	4	2	2	4					
Тема 2.7 Поняття конструктору класу. Принцип роботи та взаємодії з ним. Магічні методи в Python.	8	4	2	2	4					
Тема 2.8 Спадкування в ООП. Інкапсуляція в ООП.	8	6	2	4	2					
Тема 2.9 Введення в Tkinter. Створення вікон.	8	4	2	2	4					
Тема 2.10 Введення в віджети. Tk и ttk. Обробка подій.	10	6	2	4	4					
Тема 2.11 Основні елементи вікон в Tkinter.	10	6	2	4	4					
Тема 2.12 Позиціонування елементів вікон в Tkinter.	8	4	2	2	4					
Тема 2.13 Віджет Text в Tkinter.	8	4	2	2	4					

Тема 2.14 Віджет Treeview. Створення таблиць та дерев в Tkinter.	6	4	2	2	2					
Тема 2.15 Вікна. Стилізація вікон.	10	6	3	3	4					
Тема 2.16 Підсумкове заняття. Аналіз пройденого матеріалу	2	2	2	0	0					
Разом за змістовим модулем 2	122	72	35	37	50					
Курсова робота	30	0	0	0	30					
РАЗОМ за II семестр	152	72	35	37	50					
Всього з дисципліни	255	123	60	63	132					

5. МЕТОДИ ТА ФОРМИ НАВЧАННЯ

I. Методи організації та здійснення навчально-пізнавальної діяльності

Словесні методи (бесіда, розповідь, пояснення, лекції тощо) характерні тим, що інформацію для засвоєння здобувач освіти отримує вербальними засобами, тобто через слово.

Наочні методи - інформація для засвоєння одержується на основі сенсорно- перцептивної діяльності (демонстрування, ілюстрації, показ об'єкта, моделі).

Практичні методи. Суть їх у тому, що шляхом виконання практичних дій здобувач освіти отримує деяку інформацію, яку аналізує, робить висновок і приходять до тих знань, які необхідно засвоїти. Особливість методу в тому, що діяльність з одержання знань накладається в часі на діяльність з їх застосування, що дає винятково важливий педагогічний ефект.

II. Методи стимулювання інтересу до навчання і мотивації навчально-пізнавальної діяльності.

1. Бесіда, або діалог з аудиторією. Ставиться серія запитань, які потребують відповіді. Це дає можливість зрозуміти, чи готові здобувачі освіти сприймати новий матеріал, чи їх потрібно активізувати. Практика підказує, що здобувачі освіти ідуть на заняття не підготовлені, але коли знають, що буде опитування – готуються. Разом з тим це дає можливість виявити прогалини, що важливо не стільки для здобувача освіти, як для викладача.

2. Дискусія. Відбувається активний обмін думками. Це різновид проблемних лекцій. Проводяться ділові ігри, самостійна робота. Лекція-дискусія дає можливість охопити складний, великий за обсягом і найбільш вдалий матеріал.

3. Аналіз конкретних ситуацій. Береться конкретна ситуація з життя (професійна діяльність, соціум тощо) і вирішується різними шляхами. Сьогодні неможливо навчати здобувача освіти старими методами. Знань стало так багато, професійні навички стали настільки багатоманітними, що їх неможливо передати в повному обсязі в межах традиційних методів, шляхом ретрансляції, позбавленої емоційності.

4. Заняття з використанням техніки зворотного зв'язку. Після подачі лекції починається її обговорення. З'ясовується наскільки здобувачі освіти зрозуміли матеріал.

5. Метод «заверши фразу». Здобувач освіти може продовжувати її своїми словами, а не так як у конспекті.

6. Консультація. Для індивідуальної роботи зі здобувачами освіти використовують **пояснення**.

6. МЕТОДИ ТА ФОРМИ КОНТРОЛЮ

За місцем у навчальному процесі розрізняють **вхідний, поточний, періодичний, підсумковий види контролю.**

Вхідний контроль – використовують перед вивченням нової теми на початку семестру для з'ясування загального рівня підготовки здобувачів освіти з дисципліни, щоб передбачити організацію їх навчально-пізнавальної діяльності.

Поточний контроль – спостереження викладача за навчальною діяльністю здобувачів освіти на занятті. Метою його є отримання оперативних даних про рівень знань здобувачів освіти і якість навчальної роботи на занятті, оптимізація управління навчальним процесом.

Періодичний (тематичний) контроль – виявлення й оцінювання засвоєних на кількох попередніх заняттях знань, умінь здобувачів освіти з метою визначення, наскільки успішно вони володіють системою знань, чи відповідають ці знання програмі. Різновидом періодичного є **тематичний контроль**, що полягає у перевірці та оцінюванні знань здобувачів освіти з кожної теми і спрямований на те, щоб усі належно засвоїли кожну тему.

Підсумковий контроль здійснюється наприкінці семестру або навчального року. Підсумкову оцінку за семестр виставляють за результатами тематичного оцінювання, за рік – на основі семестрових оцінок.

Навчальні досягнення здобувачів освіти з навчальної дисципліни «Основи програмування» можуть оцінюватися за кредитно-трансферною системою ЄКТС, в основу якої покладено принцип прозорості, об'єктивності, індивідуальності та певної уніфікованості. Головне завдання при цьому – досягти найбільш ефективного та об'єктивного оцінювання, яке повинне одночасно виконувати контролюючу й мотивуючу функції.

Кожен модуль включає лекційні та лабораторні заняття, самостійну роботу.

Модульний контроль знань здобувачів освіти здійснюється через проведення аудиторних письмових контрольних робіт або комп'ютерного тестування.

Кількість балів за роботу з теоретичним матеріалом, на практичних заняттях, під час виконання самостійної роботи залежить від дотримання таких вимог:

- своєчасність виконання завдань;
- повний обсяг їх виконання;
- якість виконання навчальних завдань;
- самостійність виконання;
- творчий підхід до виконання завдань;
- ініціативність у навчальній діяльності.

Форма підсумкового контролю успішності навчання – диференційований залік – 3й семестр 2го року навчання, екзамен – 4й семестр 2го року навчання (денна форма).

7. ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ НАВЧАННЯ

Оцінювання навчальних досягнень здобувачів освіти здійснюється відповідно до «Положення про оцінювання навчальних досягнень здобувачів освіти у Білгород-Дністровському фаховому коледжі природокористування, будівництва та комп'ютерних технологій».

Формою семестрової атестації є диференційований залік – 3й семестр 2го року навчання, екзамен – 4й семестр 2го року навчання (денна форма).

Результати навчання здобувачів фахової передвищої освіти Коледжу з теоретичної та практичної підготовки можуть оцінюватись за 100-бальною шкалою, оцінкою в ЄКТС.

Відповідно рейтинг здобувача освіти із засвоєння навчальної дисципліни може складатися з рейтингу з навчальної роботи – 70 балів та рейтингу з атестації – 30 балів. Таким чином, на оцінювання засвоєння змістових модулів, на які поділяється навчальний матеріал дисципліни, передбачається 70 балів. Рейтингові оцінки із змістових модулів, як і рейтинг з атестації, теж обчислюються за 100-бальною шкалою.

Для занесення оцінок до екзаменаційної відомості, індивідуального навчального плану здобувача освіти (залікової книжки) та журналу рейтингової оцінки знань здобувача освіти його рейтинг з різних видів навчальної роботи у балах переводиться у національну та ЄКТС (Європейська кредитна трансферно-накопичувальна система) оцінки згідно з таблицею.

Відповідність результатів контролю знань за різними шкалами і критерії оцінювання

Оцінка ЄКТС	Сума балів за 100 бальною шкалою	Національна шкала (12-бальна)	Національна шкала (4-бальна)	Рівень компетентності	Критерії оцінювання
A	90 – 100 (відмінно)	12-10	відмінно	Високий рівень	Здобувач освіти виявляє особливі творчі здібності, вміє самостійно здобувати знання, без допомоги викладача знаходить та опрацьовує необхідну інформацію, вміє використовувати набуті знання і вміння для ухвалення рішень у нестандартних ситуаціях, переконливо аргументує відповіді, самостійно розкриває власні обдарування і нахили.
B	85 – 89 (дуже добре)	9-8	добре	Достатній рівень	Здобувач освіти вільно володіє вивченим обсягом матеріалу, застосовує його на практиці, вільно розв'язує вправи і задачі у стандартних ситуаціях, самостійно виправляє допущені помилки, кількість яких незначна.
C	75 – 84 (добре)	7			Здобувач освіти вміє зіставляти, узагальнювати, систематизувати інформацію під керівництвом викладача; в цілому самостійно застосовувати її на практиці; контролювати власну діяльність; виправляти помилки, серед яких є суттєві, добирати аргументи для підтвердження думок.
D	70 – 74 (задовільно)	6-5	задовільно	Середній рівень	Здобувач освіти відтворює значну частину теоретичного матеріалу, виявляє знання і розуміння основних положень; з допомогою викладача може аналізувати навчальний матеріал, виправляти помилки, серед яких є значна кількість суттєвих.
E	60 – 69 (достатньо)	4			Здобувач освіти володіє навчальним матеріалом на рівні, вищому за початковий, значну частину його відтворює на репродуктивному рівні.
FX	35 – 59 (незадовільно)	3	незадовільно	Початковий рівень	Здобувач освіти володіє матеріалом на рівні окремих фрагментів, що становлять незначну частину навчального матеріалу.
F	1 – 34 (незадовільно)	2			Здобувач освіти володіє матеріалом на елементарному рівні засвоєння, викладає його уривчастими реченнями, виявляє здатність викласти думку на елементарному рівні.
		1			Здобувач освіти володіє навчальним матеріалом на рівні елементарного розпізнавання і відтворення окремих фактів, елементів, об'єктів, що позначаються здобувачем освіти окремими словами чи реченнями.

8. МЕТОДИЧНЕ ЗАБЕЗПЕЧЕННЯ КУРСУ

1. Підручники та посібники.
2. Конспекти лекцій.
3. Лекції на електронних носіях
4. Презентації
5. Методичні вказівки до практичних робіт.
6. Матеріали для самостійного вивчення на електронних носіях.
7. Індивідуальні завдання студентів
8. Методичні вказівки до курсового проекту
9. Методичні вказівки до розрахунково-конструктивного розділу дипломного проекту
10. Матеріали з контролю знань студентів
11. Стенди та інші наглядне обладнання аудиторії

Вивчення дисципліни здобувачами освіти передбачає вміння використовувати різні інформаційні ресурси – опубліковану українську та іноземну літературу (нормативні документи, підручники, навчальні посібники, наукові періодичні та монографічні видання, словники, довідники тощо), методичну літературу та Інтернет-джерела.

9. РЕКОМЕНДОВАНІ ДЖЕРЕЛА ІНФОРМАЦІЇ

1. Навчальні матеріали: Python [Електронний ресурс] – Режим доступу до ресурсу: <http://www.matfiz.univ.kiev.ua/pages/13>.
2. Крєневич А.П. Python у прикладах і задачах. Частина 1. Структурне програмування – К.:ВПЦ "Київський Університет", 2017. – 206 с.
3. Яковенко А.В. Основи програмування. Python. Частина 1.– Київ: КПІ ім. Ігоря Сікорського, 2018. – 195 с. Електронний ресурс: <https://ela.kpi.ua/bitstream/123456789/25111/1/Python.pdf>
4. Маттеє Е. Пришвидшений курс Python. – Львів: ВСЛ, 2021. – 600 с.
5. Васильєв О. програмування в Python. Теорія та практика. – К.: Ліра-К, 2023. – 462 с.
6. Костюченко А.О. Основи програмування мовою Python: навчальний посібник. Чернігів : ФОП Баликіна С.М., 2020. 180 с.
7. Яковенко А. В. Основи програмування. Python. Частина 2 [Електронний ресурс]: Київ : КПІ ім. Ігоря Сікорського, 2018. 195 с.
8. Lutz M. Learning Python, 5th Edition. O'Reilly Media, 2013. 1643 p.
9. Бабич Т.Ю. Методичні вказівки до виконання лабораторних робіт та самостійної роботи з навчальної дисципліни «Основи програмування. Python» для здобувачів вищої освіти першого (бакалаврського) рівня та другого

(магістерського) рівня усіх освітньо-професійних програм спеціальностей НУВГП денної форми навчання. Рівне : НУВГП, 2021. 49 с.
<https://ep3.nuwm.edu.ua/20825/1/04-05-50M.pdf>

10. Беррі П. Head First. Python. Легкий для сприйняття довідник. Харків : Фабула. 2021. 624 с.

Допоміжна

11. Васильєв О. Програмування мовою Python : навч. посіб. / А. П. Бойко. – Київ : Вид-во Навчальна книга Богдан, 2019. – 504 с.

12. Мізюк О. Путівник програмування мовою Python [Електронна версія]. –Режим доступу:<https://pythonguide.rozh2sch.org.ua/>