

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ПРИКАРПАТСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ВАСИЛЯ СТЕФАНІКА**



**Факультет фізико-технічний**

**Кафедра комп'ютерної інженерії та електроніки**

**СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

**Програмування на Python**

Освітня програма: «Комп'ютерна інженерія»

Спеціальність: 123 - Комп'ютерна інженерія

Галузь знань: 12 - Інформаційні технології

Затверджено на засіданні кафедри  
комп'ютерної інженерії та електроніки  
протокол № 12 від 30 грудня 2023р.

м. Івано-Франківськ – 2023 р.

## ЗМІСТ

1. Загальна інформація
2. Анотація до курсу
3. Мета та цілі курсу
4. Компетентності
5. Результати навчання
6. Організація навчання курсу
7. Система оцінювання курсу
8. Політика курсу
9. Рекомендована література

<b>1. Загальна інформація</b>	
<b>Назва дисципліни</b>	Програмування на Python
<b>Рівень вищої освіти</b>	Перший рівень вищої освіти
<b>Викладач</b>	Голота Віктор Іванович
<b>Контактний телефон викладача</b>	(0342) 59-60-07
<b>Е-mail викладача</b>	victor.holota@pnu.edu.ua
<b>Формат дисципліни</b>	Семестровий
<b>Обсяг дисципліни</b>	3 кредити ЄКТС, 90 год.
<b>Посилання на сайт дистанційного навчання</b>	<a href="https://d-learn.pnu.edu.ua/">https://d-learn.pnu.edu.ua/</a>
<b>Консультації</b>	Відповідно до графіку консультацій, який розміщений на інформаційному сайті кафедри комп'ютерної інженерії та електроніки
<b>2. Анотація до навчальної дисципліни</b>	
<p>Дисципліна «Програмування на Python» належить до переліку вибіркових навчальних дисциплін за освітнім рівнем «бакалавр», що пропонуються в рамках циклу професійної та практичної підготовки студентів за освітньою програмою «Комп'ютерна інженерія».</p> <p>Предметом навчальної дисципліни є вивчення додаткових можливостей і бібліотек мови Python для роботи з комп'ютерною графікою, файлами, базами даних, регулярними виразами, потоками і процесами, тестуванням, реактивним програмуванням.</p>	
<b>3. Мета та цілі навчальної дисципліни</b>	
<p><u>Метою</u> вивчення навчальної дисципліни "Програмування на Python" є формування у студентів знань, вмінь та навичок розробки програм з використанням додаткових можливостей і бібліотек мови Python.</p>	
<b>4. Програмні компетентності та результати навчання</b>	
<p>Завдання вивчення дисципліни вивчення дисципліни "Програмування на Python" є отримання теоретичних знань та практичних для роботи з: комп'ютерною графікою; файлами даних; базами даних; регулярними виразами; процесами і потоками; тестуванням програмного забезпечення; реактивного програмування.</p> <p>У результаті вивчення навчальної дисципліни студент повинен знати:</p> <p>основи комп'ютерної графіки з використанням бібліотек Matplotlib, Turtle, PyOpenGL; особливості модульного, інтеграційного і системного тестування; бібліотеки для роботи з файлами, каталогами і шляхами; класи для роботи з буферизованим і небуферизованими потоками; шаблони метасимволів і регулярні вирази, модуль re; інтерфейс до баз даних DB API, модуль PyMySQL; класи і потоки для роботи з процесами і потоками; переваги і недоліки</p>	

реактивного програмування.

вміти:

розробляти програми комп'ютерної графіки з використанням графічних бібліотек; розробляти програми для модульного та інтеграційного тестування; розробляти програми для роботи з файлами і буферизованими і небуферизованими потоками; використовувати у програмах регулярні вирази; розробляти інтерфейси до баз даних; працювати з процесами і потоками та організувати їх синхронізацію; розробляти програми з реактивними властивостями.

#### **4. Компетентності**

##### **Інтегральна компетентність**

Здатність розв'язувати складні спеціалізовані задачі та практичні проблеми під час професійної діяльності в комп'ютерній галузі або навчання, що передбачає застосування теорій та методів комп'ютерної інженерії і характеризується комплексністю та невизначеністю умов

##### **Загальні компетентності**

Здатність до абстрактного мислення, аналізу і синтезу.

Здатність застосовувати знання у практичних ситуаціях

Вміння виявляти, ставити та вирішувати проблеми.

##### **Спеціальні (фахові) компетентності**

Здатність використовувати сучасні методи і мови програмування для розроблення алгоритмічного та програмного забезпечення.

Здатність створювати системне та прикладне програмне забезпечення комп'ютерних систем та мереж

#### **5. Результати навчання**

Знати новітні технології в галузі комп'ютерної інженерії.

Вміти системно мислити та застосовувати творчі здібності до формування нових ідей.

Вміти розробляти програмне забезпечення для вбудованих і розподілених застосувань, мобільних і гібридних систем, розраховувати, експлуатувати, типове для спеціальності обладнання.

Здатність використовувати типові та розробляти власні програмні продукти, орієнтовані на розв'язок задач проектування та розрахунку складових частин спеціалізованих комп'ютерних систем для оптимізації структури та конструкції досліджуваних об'єктів, підготовки необхідної технологічної документації.

Знати і розуміти наукові положення, що лежать в основі функціонування комп'ютерних засобів, систем та мереж.

Вміти розв'язувати задачі аналізу та синтезу засобів, характерних для спеціальності

#### **6. Організація навчання**

Обсяг навчальної дисципліни

Вид заняття		Загальна кількість годин		
лекції		14		
практичні заняття		16		
самостійна робота		60		
Ознаки навчальної дисципліни				
Семестр	Спеціальність	Курс (рік навчання)	Нормативний/ вибірковий	
7	G5	4	вибірковий	
Тематика навчальної дисципліни				
Тема		Кількість годин		
		лекції	практичні заняття	сам. роб
<b>Тема 1. Вступ в комп'ютерну графіку.</b> Бітова і векторна графік. Бібліотеки комп'ютерної графіки Matplotlib, Turtle, PyOpenGL.		2		4
<b>Тема 2. Тестування програм.</b> Модульне, інтеграційне і системне тестування. Розроблення програм кероване тестуванням. Середовище PyTest.		2	2	8
<b>Тема 3. Робота з файлами і потоками введення/виведення.</b> Читання і записування файлів. Класи для роботи з буферизованими і небуферизованими потоками введення/виведення. Робота з CSV і Excel файлами.		2	2	8
<b>Тема 4. Регулярні вирази.</b> Шаблони регулярних виразів. Модуль re. Об'єкт Match. Функції search(), match(), findall(), finditer(), split(), sub(), compile().		2	2	8
<b>Тема 5. Доступ до баз даних.</b> Вступ до баз даних. Мова маніпуляції даними. Інтерфейс Python DB-API. Modul PyMySQL. Робота з модулем PyMySQL. Транзакції.		2	2	8
<b>Тема 6. Процеси і потоки.</b> Паралелізм і перегони. Бібліотеки і класи для роботи з процесами. Модуль Multiprocessing. Потоки. Клас Thread. Синхронізація процесів/потоків. Синхронізаційні примітиви.		2	4	12
<b>Тема 7. Реактивне програмування.</b> Вступ в реактивне програмування. Переваги і		2	4	12

недоліки реактивного програмування. Шаблон Observer. Середовище реактивного програмування RxPy.			
<b>ЗАГ.:</b>		14	16
<b>6. Система оцінювання навчальної дисципліни</b>			
Загальна система оцінювання навчальної дисципліни	<p>Поточний контроль проводиться на всіх видах аудиторних занять. Поточний контроль може проводитись у формі усного опитування або письмового експрес-контролю на практичних заняттях та лекціях, у формі виступів здобувачів вищої освіти при обговоренні питань на семінарських заняттях, у формі комп'ютерного тестування тощо. Результати поточного контролю є основною інформацією при проведенні заліку і враховуються при визначенні підсумкової оцінки з відповідної навчальної дисципліни.</p> <p>Модульний контроль (сума балів за окремий змістовий модуль) проводиться (виставляється) на підставі оцінювання результатів знань студентів після вивчення матеріалу з логічно завершеної частини дисципліни – змістового модуля. Передбачено два змістовних модулі, кожен з яких завершується підсумковим тестуванням. Завданням модульного контролю є перевірка розуміння та засвоєння певного матеріалу (теми), вміння вирішувати конкретні ситуативні задачі, самостійно опрацювати тексти, здатності осмислювати зміст даної частини дисципліни, уміння публічно чи письмово подати певний матеріал.</p> <p>Залік – форма підсумкового контролю, що полягає в оцінюванні засвоєння здобувачем освіти навчального матеріалу з певної навчальної дисципліни, і складається зі зданих залікових змістових модулів, виконаних тестових завдань, ситуаційних робіт, опрацювання завдань робочих зошитів, тематичних рефератів, лабораторних робіт тощо, передбачених силабусом.</p> <p>Семестровий контроль у формі заліку передбачає, що підсумкова оцінка (у стобальній шкалі) з навчальної дисципліни визначається як сума оцінок за поточний контроль знань. Порядок та система оцінювання передбачається у силабусах дисципліни.</p> <p>Максимальна оцінка – 100 балів. Оцінювання здійснюється за національною та ECTS шкалами оцінювання на основі 100-бальної системи. (Див.: пункт „8.3. Види контролю” Положення про організацію освітнього процесу та розробку основних документів з організації освітнього процесу в Прикарпатському національному університеті ім. Василя Стефаника</p>		

	( <a href="https://nmv.pnu.edu.ua/нормативні документи/polozhenja/">https://nmv.pnu.edu.ua/нормативні документи/polozhenja/</a> ).
Вимоги до письмових робіт	Підсумкова робота може виконуватися у формі письмової роботи, яка містить теоретичні та/або практичні завдання і передбачає усний захист. Підсумкова робота може виконуватися у формі тестових завдань з вибором правильної відповіді
Практичні заняття	На практичних заняттях оцінюється: володіння основними поняттями і законами відповідної теми; участь у виконанні завдань, вміння самостійно розв'язувати завдання відповідної теми. Дисципліна включає 6 практичних робіт, які передбачають самопідготовку, виконання роботи на занятті, написання звіту та усний захист. При оцінюванні дисципліни враховуються бали набрані при поточному контролі. Оцінка за кожну роботу виставляється як середнє арифметичне трьох оцінок: оцінки за підготовку, виконання практичної роботи та захист (усна відповідь).
Умови допуску до підсумкового контролю	Виконання усіх запланованих програмою дисципліни форм навчальної роботи, які підлягають контрольному оцінюванню. Мінімальна кількість балів для позитивного зарахування курсу – 50 балів.
Підсумковий контроль	Форма контролю: залік. Загальні 100 балів включають: поточний контроль: захист практичних робіт: максимально 90 балів; підсумковий контроль у вигляді тестів: максимально 10 балів; додаткові бали за виконання студентських наукових робіт (наукові доповіді, тези, статті тощо): максимально 30 балів.
<b>7. Політика навчальної дисципліни</b>	
Жодні форми порушення академічної доброчесності не толеруються. Академічна доброчесність регулюється нормативною базою університету <a href="https://pnu.edu.ua/polozhennia-pro-zapobihannia-plahiatu/">https://pnu.edu.ua/polozhennia-pro-zapobihannia-plahiatu/</a> та законодавством України.	
<b>8. Рекомендована література</b>	
<b>Базова</b>	
<ol style="list-style-type: none"> <li>1. Крєневич А.П. Python у прикладах і задачах. Частина 2. Об'єктно-орієнтоване програмування. Навчальний посібник – К.: ВПЦ "Київський Університет", 2020. – 152 с</li> <li>2. Олексій Васильєв. Програмування мовою Python. – Тернопіль: "Богдан", 2019 – 504 с.</li> </ol>	

3. 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.

4. Програмування числових методів мовою Python: підруч. / А. В. Анісімов, А. Ю. Дорошенко, С. Д. Погорілий, Я. Ю. Дорогий; за ред. А. В. Анісімова. – К.: Видавничо-поліграфічний центр "Київський університет", 2014. – 640 с.

#### **Допоміжна**

5. Eric Matthes. Python Crash Course, 3rd Edition: A Hands-On, Project-Based Introduction to Programming 3rd Edition. No Starch Press, 2023. – 552 p.

6. Luciano Ramalho. "Fluent Python. Clear, Concise, and Effective Programming. 2nd Edition". O'reilly Media, 2022. – 1012 p.

7. Michal Jaworski, Tarek Ziade. Expert Python programming. Packt, 2021. – 611 p.

8. John Hant. Advanced guide to Python 3 programming. Springer, 2019, – 497.

**Викладач:**

**Голота В.І.**