

Міністерство освіти і науки України  
Державний вищий навчальний заклад  
«Прикарпатський національний університет імені Василя Стефаника»

Факультет математики та інформатики  
Кафедра диференціальних рівнянь і прикладної математики

**СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**  
ПРОГРАМУВАННЯ НА PYTHON

**Освітня програма:** Прикладна математика,  
Комп'ютерне моделювання  
та технології програмування

**Спеціальність:** 113 Прикладна математика

**Галузь знань:** 11 Математика та статистика

Затверджено на засіданні кафедри  
Протокол №1 від 31 серпня 2021 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

## 1. ЗАГАЛЬНА ІНФОРМАЦІЯ

Назва дисципліни	Програмування на Python
Викладач(-і)	Мазуренко В.В.
Контактний телефон	(0342)596027
E-mail	<a href="mailto:viktor.mazurenko@pnu.edu.ua">viktor.mazurenko@pnu.edu.ua</a>
Профайл	<a href="http://mazurenko.pnu.edu.ua">mazurenko.pnu.edu.ua</a>
Формат дисципліни	Лекції та лабораторні заняття
Обсяг дисципліни	6 кредитів
Посилання на сайт дистанційного навчання	<a href="https://classroom.google.com/c/MTY3NDk3MTYyNjY3">classroom.google.com/c/MTY3NDk3MTYyNjY3</a>
Консультації	Четвер, 15 <sup>00</sup>

## 2. АНОТАЦІЯ ДО КУРСУ

У курсі «Програмування на Python» студенти спеціальності «Прикладна математика» мають нагоду познайомитись з простою у засвоєнні і достатньо потужною для розв'язання прикладних задач мовою програмування Python, яка надає зручні високорівневі структури даних, а також простий, але ефективний підхід до об'єктно-орієнтованого програмування. Його лаконічний синтаксис і динамічна типізація разом з тим, що він є інтерпретованим і підтримує модульність, роблять Python чудовою мовою для написання сценаріїв та швидкої розробки додатків у різних областях (автоматизація задач, аналіз даних, графічний інтерфейс користувача, веб-розробка, розробка мобільних додатків, машинне навчання) і на більшості платформ (GNU/Linux, Windows, MacOS, FreeBSD, Solaris, OS/2, iOS, Android, Windows CE, PlayStation та інші).

## 3. МЕТА ТА ЦІЛІ КУРСУ

Набуття студентами загальних і фахових компетентностей з програмування для автоматизації задач у різних сферах людської діяльності. Ознайомити студентів з основними поняттями, парадигмами і базовими структурами програмування, синтаксисом мови Python; сформувані у студентів уміння і навички написання програм мовою Python для автоматизації задач різної складності на основі розробки алгоритмів їх розв'язання.

#### 4. КОМПЕТЕНТНОСТІ

Відповідно до освітньо-професійних програм «Прикладна математика» та «Комп'ютерне моделювання і технології програмування» для першого (бакалаврського) рівня вищої освіти:

- ФК04.** Здатність розробляти алгоритми та структури даних, програмні засоби та програмну документацію.
- ФК05.** Здатність проектувати бази даних, інформаційні системи та ресурси.
- ФК07.** Здатність експлуатувати та обслуговувати програмне забезпечення автоматизованих та інформаційних систем різного призначення.
- ФК08.** Здатність використовувати сучасні технології програмування та тестування програмного забезпечення.
- ФК09.** Здатність до проведення математичного і комп'ютерного моделювання, аналізу та обробки даних, обчислювального експерименту, розв'язання формалізованих задач з допомогою спеціалізованих програмних засобів.

#### 5. РЕЗУЛЬТАТИ НАВЧАННЯ

- РН09.** Будувати ефективні щодо точності обчислень, стійкості, швидкодії та витрат системних ресурсів алгоритми для чисельного дослідження математичних моделей та розв'язання практичних задач.
- РН11.** Вміти застосовувати сучасні технології програмування та розроблення програмного забезпечення, програмної реалізації чисельних і символьних алгоритмів.
- РН13.** Використовувати в практичній роботі спеціалізовані програмні продукти та програмні системи комп'ютерної математики.

#### 6. ОРГАНІЗАЦІЯ НАВЧАННЯ КУРСУ

Обсяг курсу	
Вид заняття	Загальна кількість годин
Лекції	20
Лабораторні	40
Самостійна робота	120

Ознаки курсу				
Спеціальність, освітня програма	Рівень освіти	Курс (рік навчання)	Семестр	Нормативна/вибіркова
113 Прикладна математика	Бакалавр	1 <sup>ий</sup>	1 <sup>ий</sup>	нормативна

## Тематика курсу

Тема, план	Форма заняття	Література	Завдання, год	Вага оцінки	Термін виконання
<p><b>Перше знайомство з мовою Python</b></p> <ul style="list-style-type: none"> <li>- основні парадигми програмування та їх реалізація сучасними мовами</li> <li>- поняття алгоритму і програми, основні етапи побудови програми</li> <li>- компіляція та інтерпретація програмного коду</li> <li>- про автора розробки і офіційне представлення мови Python</li> <li>- короткий опис, сфери застосування і філософія мови</li> <li>- версії інтерпретатора Python і запуск інтерпретатора в інтерактивному режимі; інтегровані середовища розробки (IDLE)</li> </ul>	лекція	[1–6, 8]	Опрацювати матеріал лекції з рекомендованою літературою, 2 ауд. год., 4 год. с. р.	–	1 <sup>ий</sup> тиждень
<p><b>Лінійна структура (послідовне виконання) програми</b></p> <ul style="list-style-type: none"> <li>- синтаксис мови Python: основні правила, ідентифікатори і ключові слова, константи і змінні, коментування програмного коду</li> <li>- поняття об'єкту, його ідентичність, тип і значення; змінювані (mutable) і незмінювані (immutable) об'єкти;</li> <li>- прості типи об'єктів (int, float, complex, str, bool), інтроспекція і перетворення типів</li> <li>- базові інструкції: пропустити (pass), присвоєння (=), введення (input), виведення (print)</li> <li>- послідовне виконання інструкцій, арифметичні операції над числами (int, float, complex) та їх пріоритет</li> </ul>	лекція	[1–6, 8]	Опрацювати матеріал лекції з рекомендованою літературою, 2 ауд. год., 4 год. с. р.	–	1 <sup>ий</sup> тиждень

- як влаштовані дійсні і комплексні числа, використання функцій модулів math і smath					
<b>Лінійні програми</b>	лабораторна робота	[7, 11, 12]	Виконати і захистити лабораторну роботу 1, 4 ауд. год., 8 год. с. р.	2	2ий тиждень
<b>Умовні керуючі конструкції</b> - логічні операції і операції відношення, їх пріоритет - логічний тип об'єктів (bool): що істина, а що хибність - галуження як керуюча конструкція: умовний оператор if...else; каскадний умовний оператор if...elif...else; тернарний умовний оператор	лекція	[1–6, 8]	Опрацювати матеріал лекції з рекомендованою літературою, 2 ауд. год., 4 год. с. р.	–	3ий тиждень
<b>Галужені програми</b>	лабораторна робота	[1, 7–11]	Виконати і захистити лабораторну роботу 2, 4 ауд. год., 8 год. с. р.	3	3ий – 4ий тижні
<b>Циклічні керуючі конструкції</b> - цикл як керуюча конструкція: цикл з передумовою while, цикл по колекції for...in, переривання і продовження циклів - арифметична прогресія як колекція range, поняття про інкремент і декремент - рекурентні співвідношення та їх системи - числові послідовності та їх границі (с. р.)	лекція	[1–6, 8]	Опрацювати матеріал лекції з рекомендованою літературою, 2 ауд. год., 4 год. с. р.	–	4ий тиждень
<b>Циклічні програми</b>	лабораторна робота	[1, 7–11]	Виконати і захистити лабораторну роботу 3, 6 ауд. год., 8 год. с. р.	4	5ий – 6ий тижні
<b>Списки і кортежі</b> - списки (list): створення, індексація, зрізи, змінюваність списків - операції над списками: конкатенація і зчеплення,	лекція	[1–6, 8]	Опрацювати матеріал лекції з рекомендованою літературою, 2 ауд. год., 4 год. с. р.	–	6ий тиждень

<p>перевірка приналежності до списку, аналіз списку, модифікація списку</p> <ul style="list-style-type: none"> <li>- введення списків з клавіатури, функція <code>map()</code></li> <li>- обхід списку: за індексами, за елементами, через функції <code>enumerate()</code> і <code>zip()</code>; матриці як списки зі списків</li> <li>- кортежі (<code>tuple</code>): створення, індексація, зрізи, незмінюваність кортежів</li> <li>- операції над кортежами: аналогічні до операцій над списками, пакування і розпакування кортежів, обмін значеннями змінних</li> <li>- генератори послідовностей (зокрема списків і кортежів), методи <code>iter()</code> і <code>next()</code>, функція <code>sum()</code> (с. р.)</li> </ul>					
<p><b>Застосування списків і кортежів</b></p>	лабораторна робота	[1, 7–11]	Виконати і захистити лабораторну роботу 4, 4 ауд. год., 8 год. с. р.	3	7ий тиждень
<p><b>Символи і рядки</b></p> <ul style="list-style-type: none"> <li>- символи і таблиці кодування, спеціальні (екрановані) символи, операції із символами</li> <li>- рядок як кортеж символів, індексація, зрізи, довжина рядка, незмінюваність рядків</li> <li>- операції над рядками: конкатенація і зчеплення, перевірка приналежності до рядка, пошук і заміна, аналіз рядка, модифікація рядка, розбиття і побудова рядка</li> <li>- порівняння і форматування рядків, метод <code>format()</code> і <code>f-рядки</code> (с. р.)</li> </ul>	лекція	[1–6, 8]	Опрацювати матеріал лекції з рекомендованою літературою, 2 ауд. год., 4 год. с. р.	–	8ий тиждень
<p><b>Застосування символів і рядків</b></p>	лабораторна робота	[1, 7–11]	Виконати і захистити лабораторну роботу 5, 4 ауд. год., 8 год. с. р.	3	8ий – 9ий тижні

<p><b>Словники і множини</b></p> <ul style="list-style-type: none"> <li>- словники (dict): створення і доступ до елементів, невпорядкованість і незмінюваність словників</li> <li>- операції над словниками: перевірка приналежності до словника, аналіз словника, модифікація списку</li> <li>- обхід словника: за ключами, за значеннями, за парами «ключ-значення»</li> <li>- множини (set): створення, індексація, зрізи, невпорядкованість і змінюваність множин</li> <li>- операції над множинами: операції відношення; перетин, об'єднання, різниця і симетрична різниця; додавання і видалення елементів</li> <li>- незмінювані множини (frozenset) (с. р.)</li> </ul>	лекція	[1–6, 8]	Опрацювати матеріал лекції з рекомендованою літературою, 2 ауд. год., 4 год. с. р.	–	9ий тиждень
<p><b>Застосування словників і множин</b></p>	лабораторна робота	[1, 7–11]	Виконати і захистити лабораторну роботу 6, 4 ауд. год., 8 год. с. р.	3	10ий тиждень
<p><b>Функції</b></p> <ul style="list-style-type: none"> <li>- поняття підпрограми і функції, визначення і виклик функцій, повернення кількох значень</li> <li>- аргументи (параметри) функції: формальні і фактичні, передача аргументів: позиційні та іменовані (ключові) аргументи; типові (за замовчуванням) значення аргументів</li> <li>- функції з довільною кількістю аргументів, області видимості: локальні і глобальні змінні</li> <li>- поняття про функційний тип даних, анонімні (lambda-функції)</li> <li>- рекурсія і рекурсивні функції, генератор-функції</li> <li>- декоратори для функцій: означення, загальний опис, вкладені декоратори (с. р.)</li> </ul>	лекція	[1–6, 8]	Опрацювати матеріал лекції з рекомендованою літературою, 2 ауд. год., 4 год. с. р.		11ий тиждень

<b>Створення функцій</b>	лабораторна робота	[1, 7–11]	Виконати і захистити лабораторну роботу 7, 4 ауд. год., 8 год. с. р.	4	11 <sup>ий</sup> – 12 <sup>ий</sup> тижні
<b>Обробка помилок і винятків</b> - помилки у програмному кодї та їх типи: синтаксичні, логічні і помилки виконання - виняткові ситуації, обробка винятків з допомогою try...except, стандартні класи винятків, створення та ініціювання користувацьких винятків - менеджер контексту with...as.	лекція	[1–6, 8]	Опрацювати матеріал лекції з рекомендованою літературою, 2 ауд. год., 4 год. с. р.		12 <sup>ий</sup> тиждень
<b>Файли і модулі</b> - файл і його ознаки, типи файлів - етапи роботи з файлом: відкриття, читання і запис, закриття; текстові файли - бінарні файли і серелізатори - робота з файловою системою - імпорт і використання модулів - головний модуль програми, розташування модулів у файловій системі - пакет і його структура (с. р.)	лекція	[1–6, 8]	Опрацювати матеріал лекції з рекомендованою літературою, 2 ауд. год., 4 год. с. р.		13 <sup>ий</sup> тиждень
<b>Робота з файлами</b>	лабораторна робота	[1–6, 8]	Виконати і захистити лабораторну роботу 8, 4 ауд. год., 8 год. с. р.	3	13 <sup>ий</sup> – 14 <sup>ий</sup> тижні
<b>Тематичний контроль</b>	тест	[1–6, 8–9]	Підгот. до тесту, 12 год. с. р. Індивід. завдання, 2 ауд. год.	25	15 <sup>ий</sup> тиждень
<b>Підсумковий контроль</b>	іспит	[1–11]	Підгот. до іспиту, 10 год. с. р. Індивід. завдання, 2 ауд. год.	50	16 <sup>ий</sup> тиждень

## 7. СИСТЕМА ОЦІНЮВАННЯ КУРСУ

Загальна система оцінювання	Підсумкова оцінка з дисципліни у відповідному семестрі є сумою оцінок за кожен з таких видів робіт: аудиторна (активна робота на лекціях) і самостійна (опрацювання окремих тем) роботи, лабораторні роботи (виконання і захист), тематичний контроль (тест) і підсумковий контроль (іспит). Підсумкова оцінка визначається відповідно до поданої нижче таблиці оцінювання за різними шкалами (100-бальна, ECTS, національна).
Лабораторні роботи	Максимальна оцінка за вчасно виконані і захищені лабораторні роботи становить 25 балів.
Самостійна робота	Максимальна оцінка за активну і змістовну аудиторну роботу та самостійну роботу за окремими темами становить 5 балів (додатково).
Тематичний контроль	Тест з програмування на Python містить 25 завдань закритого/відкритого типу на знання парадигм програмування, на розуміння синтаксису і базових конструкцій мови, на вміння працювати з простими типами даних і структурами даних, на застосування функцій зі стандартних модулів і побудову власних функцій, на тестування програмного коду і обробку помилок. Максимальна оцінка за тест становить 25 балів.
Умови допуску до підсумкового контролю	Загальна кількість балів за всі види робіт впродовж семестру становить не менше 25 балів.
Підсумковий контроль	Кожен варіант екзаменаційного білета складається з 2-х частин: тест на знання теоретичних основ програмування мовою Python та написання програм на Python для вирішення 2-х практичних задач. Максимальна оцінка за іспит становить 50 балів.

### ШКАЛА ОЦІНЮВАННЯ: НАЦІОНАЛЬНА ТА ECTS

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою	
		для екзамену, курсового проекту (роботи), практики	для заліку
90 – 100	<b>A</b>	відмінно	зараховано
80 – 89	<b>B</b>	добре	
70 – 79	<b>C</b>		
60 – 69	<b>D</b>	задовільно	
50 – 59	<b>E</b>		
26 – 49	<b>FX</b>	незадовільно з можливістю повторного складання	не зараховано з можливістю повторного складання
0-25	<b>F</b>	незадовільно з обов'язковим повторним вивченням дисципліни	не зараховано з обов'язковим повторним вивченням дисципліни

## 8. ПОЛІТИКА КУРСУ

Усі види навчальної роботи слід виконувати вчасно, щоб зберегти загальний темп курсу, котрий сприяє ефективному засвоєнню матеріалу без шкоди здоров'ю. Наслідками пропущених занять без поважних причин, зазвичай, стають додаткові завдання для самостійної роботи.

При проходженні курсу вітаються комунікативність, активність, креативність, самостійність. Плагіат та інші види академічної недоброчесності не принесуть користі, тому є недоречними.

## 9. РЕКОМЕНДОВАНА ЛІТЕРАТУРА

1. Крєневич А.П. Python у прикладах і задачах. Ч. 1. Структурне програмування: навч. посібн. – К.: ВПЦ "КУ", 2017. – 206 с. – Режим доступу: <http://www.matfiz.univ.kiev.ua/uploads/books/python2017.pdf>
2. Мізюк О. Путівник мовою програмування Python [Ел. ресурс] – 2019. – Режим доступу: <https://pythonguide.rozh2sch.org.ua/>
3. Яковенко А.В. Основи програмування. Python. Частина 1 [Ел. ресурс]. – К.: КПІ ім. Ігоря Сікорського, 2018. – 195 с. – Режим доступу: <https://ela.kpi.ua/bitstream/123456789/25111/1/Python.pdf>
4. Swaroop Ch. A byte of Python – Укус Питона [Ел. ресурс] / пер. В. Смоляр. – 2013. – 151 с. – Режим доступу: <https://wombat.org.ua/AByteOfPython/toc.html>
5. Severance Ch. R. Python for Everybody. – 2009. – 245 р. – Режим доступу: [http://do1.dr-chuck.com/pythonlearn/EN\\_us/pythonlearn.pdf](http://do1.dr-chuck.com/pythonlearn/EN_us/pythonlearn.pdf)
6. Мусин Д. Самоучитель Python. Выпуск 0.2. – 2017. – 149 с. – Режим доступа: <https://docplayer.com/25829012-Samouchitel-python-vypusk-0-2-dmitriy-musin.html>
7. Мізюк О. Практикум з програмування мовою Python [Електронний ресурс] – 2019. – Режим доступу: <https://pythonexercises.rozh2sch.org.ua/>
8. Васильев А. Н. Python на примерах. Практический курс по программированию. - СПб.: Наука и Техника, 2016. - 432 с. – Режим доступа: [http://www.dut.edu.ua/uploads/l\\_552\\_54601790.pdf](http://www.dut.edu.ua/uploads/l_552_54601790.pdf)
9. Лутц М. Python. Карманный справочник: Пер. с англ. – 5-е изд. – М.: Вильямс, 2015. – 320 с.
10. The Python Tutorial [Електронний ресурс] – Режим доступу: <https://docs.python.org/3/tutorial/index.html>.
11. E-Olymp [Електронний ресурс] – Режим доступу: <https://e-olymp.com>

**Викладач** Мазуренко В.В.