



Силабус освітнього компонента Програма навчальної дисципліни



Програмування

Шифр та назва спеціальності

105 – Прикладна фізика та наноматеріали

Інститут

ННІ Комп'ютерного моделювання, прикладної фізики та математики

Освітня програма

Прикладна фізика та наноматеріали для енергетики, медицини, радіоелектроніки та телекомунікацій

Кафедра

Радіоелектроніка (164)

Рівень освіти

Бакалавр

Тип дисципліни

Спеціальна (фахова), Обов'язкова

Семестр

1, 2

Мова викладання

Українська,

Викладачі, розробники

**Караман Дмитро Григорович**

dmytro.karaman@khpi.edu.ua

старший викладач кафедри "Автоматика та управління в технічних системах", НТУ "ХПІ"

Досвід роботи – 14 років. Автор та співавтор понад 50 наукових та навчально-методичних праць. Основні професійні та наукові інтереси: розробка комп'ютеризованих засобів автоматизації та телекомунікаційних систем з використанням сучасних технологій та радіоелектронних компонентів. До сфери інтересів також входить розробка та реалізація прикладного програмного забезпечення з використанням мов програмування високого рівня.

[Детальніше про викладача на сайті кафедри](#)

Загальна інформація

Анотація

Курс "Програмування" призначений для надання студентам фундаментальних знань з області інформатики та введення у світ програмування з використанням потужного сучасного інструменту — мови програмування Python. В процесі проходження курсу навчання студенти ознайомлюються з базовими принципами побудови обчислювальних систем і архітектурою комп'ютера, основними поняттями про інформацію, способами її отримання, обробки та подання. Студенти навчаються створювати програми для розв'язання практичних задач та аналізу даних, розвиваючи не лише технічні навички, але й креативне мислення для успішного вирішення завдань в сучасному цифровому середовищі. Цей курс є необхідним пререквізитом для опанування наступних курсів, в яких обов'язковою умовою є використання комп'ютерної техніки та створення прикладного програмного забезпечення.

Мета та цілі дисципліни

Ознайомлення з основними принципами організації та роботи обчислювальних систем. Вивчення способів подання, обробки та аналізу даних. Оволодіння теоретичними знаннями та практичними

навичками створення прикладних програм, які здатні вирішити необхідні задачі, отримання ключових навичок алгоритмізації задач, отримання та підготовки необхідних даних, формування та виведення результатів.

Формат занять

Лекції, лабораторні та практичні роботи, консультації. Передбачається самостійне опрацювання лекційного та додаткового теоретичного матеріалу, виконання самостійних практичних завдань. Проводиться проміжний модульний контроль у вигляді письмових контрольних робіт або електронного тестування. Підсумковий контроль – іспит. Учбовий матеріал надається в електронному вигляді, що дозволяє проведення занять як у аудиторній, так і у дистанційній формі.

Компетентності

Вивчення даної дисципліни забезпечує формування у студентів програмних компетентностей ЗК1, ЗК5, ЗК9, СК2, СК5 та СК7 згідно освітньої програми, а саме: здатність застосовувати отримані знання у практичних ситуаціях, навички використання інформаційних і комунікаційних технологій, здатність працювати автономно, здатність використовувати комп'ютерні технології та програмування для теоретичного дослідження та математичного моделювання в професійній діяльності, брати участь у плануванні і виконанні експериментів та лабораторних досліджень властивостей фізичних систем, фізичних явищ і процесів, обробленні й презентації їхніх результатів, а також здатність до постійного розвитку компетентностей у сфері прикладної фізики, інженерії та комп'ютерних технологій.

Результати навчання

Вивчення даної дисципліни передбачає досягнення програмних результатів навчання Р04, Р07 та Р09 згідно освітньої програми, а саме: застосовувати фізичні, математичні та комп'ютерні моделі для дослідження фізичних явищ, розробки приладів і наукоємних технологій, класифікувати, аналізувати та інтерпретувати науково-технічну інформацію в галузі прикладної фізики, а також Презентувати результати досліджень і розробок фахівцям і нефахівцям, аргументувати власну позицію.

Обсяг дисципліни

Навчання дисципліни розподілено на два семестри. Загальний обсяг дисципліни 300 год. (10 кредитів ECTS). В першому семестрі загальний обсяг складає 120 годин (4 кредити ECTS): лекції – 32 год., лабораторні роботи – 16 год., самостійна робота – 72 год. У другому семестрі загальний обсяг складає 180 годин (6 кредитів ECTS): лекції – 32 год., лабораторні роботи – 32 год., практичні заняття – 16 год., самостійна робота – 100 год.

Передумови вивчення дисципліни (пререквізити)

Студент повинен мати базові навички та знання користувача ПК, вміти здійснювати інформаційний пошук матеріалів за заданою темою, бажано володіння англійською мовою. Для вивчення дисципліни є достатніми знання з математики, фізики та інформатики на рівні середньої освіти.

Особливості дисципліни, методи та технології навчання

Усі навчальні матеріали надаються в електронному вигляді, що дозволяє проведення занять як у аудиторній, так і у дистанційній формі. Навчальні матеріали адаптовано до проведення занять з використанням LMS Microsoft Teams або Google Classroom. Для виконання лабораторних та практичних робіт необхідно використовувати персональний комп'ютер з ОС Windows або Linux (від 2 ГБ ОЗП, 1 ГБ дискового простору, доступ в Інтернет) з встановленим інтерпретатором мови Python версії не нижче 3.8 а також додатковими засобами автоматизованої розробки (редакторами програмного коду) за бажанням студента.

Програма навчальної дисципліни

Теми лекційних занять

Тема 1. Вступ

Роль програмування у сучасному світі. Необхідні знання та навички для програміста. Правильний вибір мови програмування: чому саме Python? Де і як застосовується мова Python (сфери науки та практичної діяльності, організації, професії).

Тема 2. Базові відомості про мову програмування Python

Історія виникнення та розвитку. Особливості та визначні сторони. Слабкі сторони та недоліки (а також як їх подолати). Середовище та необхідні технічні передумови.

Тема 3. Способи використання інтерпретатора та інструментів

Інтерактивний/діалоговий режим роботи з інтерпретатором (REPL). Класичний режим роботи через передавання файлу з кодом на виконання інтерпретатору. Розробницький режим роботи з кодом в середовищі проектування в режимі покрокового відлагодження.

Тема 4. Основи синтаксису Python

Регістрозалежність, чутливість до відступів. Ключові (зарезервовані) слова, вимоги до ідентифікаторів (назв користувацьких об'єктів та структур даних). Літерали та класи простих типів даних. Синтаксис коментарів.

Тема 5. Змінні

Об'єктна природа змінних та її ілюстрація. Змінні скалярних типів даних: булевих, чисел та рядків. Складені структури даних: списки, кортежі, множини, словники. Поняття про змінні та незмінні об'єкти в Python. Оператори: перелік (арифметичні, присвоювання, порівняння, логічні та бітові, спеціальні: `is` та `in`), пріоритетність, особливості застосування до літералів різних типів.

Тема 6. Рядки

Варіанти запису (різні лапки, багаторядкові літерали) та створення змінних рядкового типу. Робота з рядками: звернення до окремих елементів та діапазонів елементів (slicing). Операції над рядками. Спеціальні символи (екранування). Методи, які можна використовувати для роботи з рядками. Способи форматування рядків, включення значень інших змінних.

Тема 7. Списки (list) та кортежі (tuple)

Об'єктна природа списків, методи та функції для списків. Способи створення списку, доступ до елементів списку. Додавання, зміна та видалення елементів списку. Ітеративний характер списку та спискові вирази-включення. Сортування списку, створення копії списку, об'єднання та розділення списків. Кортежі: відмінність від списків. Об'єктна природа кортежів, методи та функції для кортежів. Створення кортежів. Запакування та розпакування кортежів. Доступ до елементів кортежу. Обхід елементів кортежу у циклі.

Тема 8. Множини (set) та словники (dictionary)

Особливості та відмінні риси множин. Об'єктна природа множин, методи та функції над множинами. Способи звернення до елементів множини. Додавання та видалення елементів множини. Обхід елементів множини за допомогою циклу. Операції над множинами. Особливість словників (відмінні риси від інших структур даних). Об'єктна природа словників. Способи створення словників, доступ до елементів словнику. Додавання, зміна та видалення елементів словнику. Робота з словником: циклічний обхід, створення копії, очищення та видалення.

Тема 9. Конструкції керування процесом виконання коду

Різновиди умовного розгалуження виконання коду та цикли. Важливість відступів (та випадки, коли ними можна знехтувати) Умвне розгалуження за допомогою конструкції `if..elif..else`. Розгалуження за допомогою конструкції `match..case`. Організація циклів за допомогою `for` - робота з ітеративними об'єктами. Класичне використання разом з функцією `range()`. Додаткові способи керування виконанням циклів: ключові слова `break`, `continue`, `pass`. Блок `else` в конструкції `for`. Організація циклів за допомогою `while` - циклічне виконання, доки не буде виконано умову. Додаткові способи керування виконанням циклів: ключові слова `break`, `continue`, `pass`. Блок `else` в конструкції `while`.

Тема 10. Функції у мові Python

Оголошення та використання. Способи передавання аргументів у функцію. Повернення результатів. Рекурсія. Зони видимості змінних: локальні та глобальні. Вбудовані функції. Отримання допомоги та відомостей про функцію. Створення та використання власних функцій.

Тема 11. Модулі та пакети у мові Python

Підключення та використання модулів у складі мови Python. Способи підключення. Виклик функцій та застосування об'єктів імпортованого модулю. Отримання допомоги з використання модулів та їх можливостей. Вбудовані стандартні модулі os, sys, math, random, datetime. Створення та використання власного модулю. Створення власного пакету.

Тема 12. Вбудовані типи помилок та обробка виключних випадків

Вбудовані типи помилок, які розпізнає інтерпретатор при виконанні користувачького коду. Конструкція перехоплення та обробки помилок у програмах. Додаткові можливості за рахунок використання гілок else та finally. Примусова генерація помилки за допомогою raise.

Тема 13. Робота з файлами в Python

Базові відомості про роботу з файлами в Python. Текстові файли. Читання та запис даних. Розташування файлів, робочий каталог та робота з файловою системою ПК. Робота з файлами у бінарному форматі. Робота зі спеціальними форматами файлів.

Тема 14. Налаштування роботи програми через передавання параметрів

Опрацювання параметрів, які передаються до модулю Python при запуску файлу програми на виконання. Деякі прийоми по роботі з командною консоллю операційної системи.

Тема 15. Об'єктно-орієнтоване програмування в Python

Загальне розуміння про об'єктно-орієнтоване програмування. Визначення понять: клас, об'єкт, спадкування, інкапсуляція, поліморфізм. Знайомство з об'єктами у Python на прикладі вбудованих класів. Створення, опис та використання об'єктів у мові Python.

Тема 16. Створення графічного інтерфейсу для програм на Python

Важливість та зручність графічного інтерфейсу для прикладних програм. Огляд різних бібліотек для створення графічного інтерфейсу у Python (Tkinter, PyQt, Kivy). Пакет Tk (Tkinter). Основні віджети та контейнери. Налаштування функціоналу та вигляду елементів інтерфейсу.

Тема 17. Створення програм на Python з використанням мережевих інтерфейсів.

Поняття сокетів та взаємодії різних програм з їх використанням. Розробка розподіленого прикладного програмного забезпечення з використанням взаємодії через сокети. Бібліотека sockets.

Тема 18. Веб-програмування на Python

Створення веб-застосунків, використання можливостей з веб-програмування для створення графічного інтерфейсу для власних застосунків за допомогою веб-браузера.

Тема 19. Багатопоточне програмування у Python

Визначення та принципи роботи з потоками. Створення та управління виконанням потоків. Бібліотека threading та її використання для багатопоточного виконання програм у Python.

Тема 20. Використання Python для роботи з базами даних

Використання вбудованого модулю sqlite3 для створення, наповнення, підтримки файлової бази даних SQLite. Способи підключення до серверних БД. Робота з базами даних типу No-SQL.

Тема 21. Використання можливостей Python для обробки даних

Статистична обробка даних (Data Science) та методи обробки великих об'ємів даних (Big Data) за допомогою Python. Пакети numpy, scipy та matplotlib. (Використання Jupyter Notebook та Google Colab).

Тема 22. Використання Python для програмування вбудованих систем (embedded systems)

Способи створення та запуску програм на мові Python на вбудованих пристроях. Використання Python на платформі Raspberry Pi. Спеціальні реалізації мови Python: MicroPython та CircuitPython для роботи на мікроконтролерних платформах для вбудованих систем (ESP8266, ESP32, Raspberry Pi Pico)

Теми практичних занять

Тема 1. Архітектура та загальні принципи роботи обчислювальних систем

Мови програмування. Введення, обробка та виведення інформації. Системи числення та взаємні переведення. Формати подання числових даних. Формати подання символічних/текстових даних.

Тема 2. Поняття алгоритму. Способи подання та відображення алгоритмів

Алгоритми простих дій. Умовне розгалуження та циклічне виконання дій. Комбінування простих дій для отримання складних алгоритмів. Основні алгоритмічні задачі у програмуванні. Задачі пошуку, сортування, матричних операцій. Рекурсивні алгоритми.

Тема 3. Основи архітектури програмного забезпечення та сучасні методології його проектування

Різновиди архітектур: монолітна, слоїста, багаторівнева, сервіс-орієнтована. Залежність архітектури від призначення ПЗ, середовища виконання, структури команди розробки та способу менеджменту. Різновиди методологій: каскадна ("водоспад"), інкрементальна, ітеративна, гнучка (agile), спіральна та ін. Залежність методології від інших аспектів розробки.

Тема 4. Системи контролю початкового коду програмних продуктів

Зберігання, резервування та контроль версій. Основні поняття та терміни. Приклади практичного застосування. Система Git та ресурс GitHub — відкриті та безкоштовні системи контролю версій ПЗ. Документування початкового коду ПЗ. Коментування як найпростіший спосіб документування ПЗ. Стандарти та загально прийняті практики документування. Системи автоматичного складання документації за початковим кодом.

Тема 5. Створення безпечного коду та функціональна перевірка розробленого ПЗ

Способи попередження помилок та конструкції для проектування безпечного коду. Методи тестування та можливості мови Python для організації перевірки створеного ПЗ.

Тема 6. Мережеві технології та мережеві інтерфейси

Принципи взаємодії програмних застосунків через сокети. Інтерфейс взаємодії програм через комп'ютерні мережі. Мережа Web та формування запитів. Взаємодія за допомогою протоколу HTTP.

Тема 7. Принципи організації та роботи багатопоточних програм

Визначення задач, які можливо розділяти на паралельні потоки. Способи побудови багатопоточного ПЗ. Можливості мови Python для організації багатопоточних застосунків.

Тема 8. Вступ до програмованої електроніки та вбудованих систем

Особливості розробки ПЗ для вбудованих систем. Практичні приклади використання мови Python для розробки застосунків для вбудованих систем.

Теми лабораторних робіт

Тема 1. Робота з літералами різних типів

Використовуючи інтерактивний режим, виконати набір дій з літералами різного типу: арифметичні дії, маніпуляції з рядками, взаємні перетворення типів.

Тема 2. Робота зі змінними

Створення та використання змінних. Дослідження природи змінних. Дослідження роботи операторів над змінними.

Тема 3. Вивчення методів роботи з рядками

Практичне дослідження операцій з рядками, фрагментування, застосування методів об'єкту рядка. Дослідження способів форматування рядків з використанням значень змінних іншого типу.

Тема 4. Робота зі списками та кортежами

Створення списків, зміна та оновлення даних, використання даних зі списків. Видалення списків. Створення та використання кортежів. Різниця між списками та кортежами.

Тема 5. Робота зі словниками та множинами

Створення словників, зміна та оновлення даних, використання даних зі словників. Видалення словників. Створення та використання множин. Різниця між словниками, множинами та іншими колекційними структурами даних.

Тема 6. Написання програм з циклами та розгалуженнями

Дослідження механізму роботи конструкцій циклів у мові Python. Створення та використання циклів для вирішення запропонованих завдань.

Тема 7. Створення та використання власних функцій

Створення та використання функцій. Вивчення способів передачі параметрів та зон видимості змінних. Вирішення запропонованих завдань за допомогою використання функцій.

Тема 8. Створення та використання власного модулю

Об'єднання створеного функціоналу у модулі. Вивчення особливостей їх створення та використання. Створення та використання власного модулю для вирішення поставлених завдань.

Тема 9. Створення стійкого та безпечного коду

Вивчення вбудованого механізму перехоплення та обробки виключних ситуацій у програмному коді. Використання цього механізму для забезпечення стабільності та керованості виконання коду у запропонованих завданнях.

Тема 10. Робота з файлами користувацьких даних

Вивчення методів та конструкцій по роботі з файлами у мові Python. Виконання завдань, які потребують завантаження початкових даних з файлів та збереження результатів роботи у файли.

Тема 11. Використання параметрів запуску програм

Способи взаємодії з операційною системою та з користувачем через механізми взаємодії з ОС. Забезпечення гнучкості роботи власного ПЗ через використання параметрів запуску на виконання.

Тема 12. Створення та використання власного класу

Засвоєння принципів побудови об'єктно-орієнтованого ПЗ. Вирішення поставлених завдань за рахунок використання класів та об'єктів у мові програмування Python.

Тема 13. Створення програми з графічним інтерфейсом

Покращення зручності взаємодії з користувачем через використання засобів графічного інтерфейсу. Засвоєння вбудованих можливостей створення програм з графічним інтерфейсом у Python.

Тема 14. Створення програм, які взаємодіють через мережевий інтерфейс

Вивчення та засвоєння принципів взаємодії програм через механізм сокетів. Практичне використання ізольованих застосунків, які взаємодіють через сокети.

Тема 15. Створення веб-застосунку

Використання веб-технологій для створення прикладного ПЗ. Можливості мови Python для створення та розгортання веб-застосунків, їх практичне застосування.

Тема 16. Дослідження можливостей багатопоточного програмування

Вивчення можливостей мови Python для створення та використання багатопотокового виконання задач у ПЗ. Створення прикладу багатопотокового застосунку.

Тема 17. Робота з базою даних

Вивчення та використання можливостей мови Python для роботи з базами даних. Створення застосунку, який може отримувати та зберігати інформацію з бази даних.

Тема 18. Обробка та візуалізація даних

Вивчення можливостей мови Python для математичного та статистичного опрацювання великих об'ємів даних. Практичний приклад обробки даних та наочної візуалізації результатів у вигляді графіків та діаграм.

Тема 19. Створення програм для вбудованих систем

Використання мови Python для розробки та виконання програм на базі вбудованих систем та мікроконтролерних пристроїв. Практичний приклад розробки вбудованого ПЗ.

Тема 20. Створення чат-боту для соціальної мережі

Використання мови Python для взаємодії з користувачем через можливості соціальних мереж та месенджерів. Взаємодія з розробленим прикладним програмним забезпеченням.

Самостійна робота

Успішне засвоєння дисципліни потребує від студента виділення додаткового часу на самостійне опрацювання лекційного та додаткового теоретичного матеріалу, роботу з джерелами інформації, доопрацювання та оформлення звітів до лабораторних робіт, а також на виконання індивідуальних комплексних проектів у вигляді розрахункових (розрахунково-графічних) завдань. В кожному семестрі студент повинен виконати по одному комплексному завданню, яке передбачає застосування отриманих теоретичних знань та практичних навичок. Перелік тем завдань оголошується викладачем на початку семестру. Кожне завдання сформовано таким чином, аби на його виконання було потрібно не більше 10 годин. Результати завдання оформлюються у вигляді письмового звіту (в електронній формі) відповідно до вимог, прийнятих у НТУ "ХПІ". Готове завдання студент має відправити на перевірку не пізніше трьох тижнів до закінчення семестру. Оцінка за завдання виставляється за прийнятою шкалою оцінювання у відповідності до критеріїв, які наведено нижче. Ця оцінка враховується при формуванні підсумкової оцінки та може вплинути на оцінку при складанні іспиту.

Література та навчальні матеріали

1. Програмування в Python. Теорія і практика / Олексій Васильєв. – Вид-во: Ліра-К, 2023. – 462 с. ISBN : 9786175205136
2. Head First Python. Легкий для сприйняття довідник / Пол Беррі. – Вид-во: Фабула, 2021. – 624 с. ISBN : 9786175220191
3. Пришвидшений курс Python. Практичний, проектно-орієнтований вступ до програмування / Ерік Маттес. – Видавництво Старого Лева, 2021. – 600 с. ISBN : 9786176798538
4. Програмування мовою Python / Олексій Васильєв. – Вид-во: Навчальна книга – Богдан, 2019. – 504 с.
5. Структурне програмування : методичні вказівки до виконання індивідуальних завдань з курсу «Технологія програмування» для студентів всіх рівнів та форм навчання спеціальностей 113 «Прикладна математика» та 122 «Комп'ютерні науки» / уклад. О. О. Ларін, М. І. Шаповалова. — Харків : НТУ «ХПІ». 2020. — 48 с.
6. Алгоритми і структури даних: практикум: навч. посіб. / Н. К. Стратієнко, М. Д. Годлевський, І. О. Бородіна. – Харків: НТУ «ХПІ», 2017. – 224 с. ISBN 978-617-05-0247-6
7. Інформатика. Програмування на Python [Електронний ресурс] : навч. посіб. для здобувачів ступеня бакалавра за освіт. програмою «Конструювання та дизайн машин» спец. 131«Прикладна механіка» / О.А. Івановський, В.С. Парненко, ; КПІ ім. Ігоря Сікорського. – Електрон. текст. дані (1 файл: 3.6 Мбайт). – Київ: КПІ ім. Ігоря Сікорського, 2023. – 239 с.
8. Підручник з Python. Офіційна документація Python. URL: <https://docs.python.org/uk/3/tutorial/index.html>
9. Путівник мовою програмування Python. Вивчення основ програмування для початківців: електронний підручник / Олександр Мізюк (gtmizyuk@gmail.com) 03.12.2020. URL: <https://pythonguide.rozh2sch.org.ua/>

Система оцінювання

Критерії оцінювання успішності студента та розподіл балів

Підсумкова оцінка формується шляхом підрахунку середнього зваженого балу на базі оцінок за виконання лабораторних робіт, індивідуальних самостійних завдань та робіт проміжного (модульного) контролю. Якщо студент не погоджується з рейтинговою оцінкою, або було виконано недостатньо завдань для формування підсумкової оцінки (менш ніж 70%), студент має отримати оцінку шляхом складання іспиту. На іспиті студент має дати відповідь на декілька (2-3) теоретичних питань за різними темами курсу та виконати практичне завдання. Оцінки виставляються за наведеною шкалою та у відповідності до критеріїв, які доступні за [посиланням](#).

Шкала оцінювання

Сума балів	Національна оцінка	ECTS
90–100	Відмінно	A
82–89	Добре	B
75–81	Добре	C
64–74	Задовільно	D
60–63	Задовільно	E
35–59	Незадовільно (потрібне додаткове вивчення)	FX
1–34	Незадовільно (потрібне повторне вивчення)	F

Норми академічної етики і політика курсу

Студент повинен дотримуватися «Кодексу етики академічних взаємовідносин та доброчесності НТУ «ХПІ»: виявляти дисциплінованість, вихованість, доброзичливість, чесність, відповідальність. Конфліктні ситуації повинні відкрито обговорюватися в навчальних групах з викладачем, а при неможливості вирішення конфлікту – доводитися до відома співробітників дирекції інституту.

Нормативно-правове забезпечення впровадження принципів академічної доброчесності НТУ «ХПІ» розміщено на сайті: <http://blogs.kpi.kharkov.ua/v2/nv/akademichna-dobrochesnist/>

Погодження
Силабус погоджено



Дата погодження, підпис
01.09.2023 р.

Завідувач кафедри
Наталія КУЗЬМЕНКО



Дата погодження, підпис
01.09.2023 р.

Гарант ОП
Сергій КОЗЛОВ