

Національний аерокосмічний університет ім. М. Є. Жуковського
«Харківський авіаційний інститут»

Кафедра аерокосмічних радіоелектронних систем (№ 501)



РОБОЧА ПРОГРАМА ОBOB'ЯЗКОВОЇ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Основи програмування

(назва навчальної дисципліни)

Галузь знань: 17 «Електроніка та телекомунікації»
(шифр і найменування галузі знань)

Спеціальність: 172 «Телекомунікації та радіотехніка»
(код та найменування спеціальності)

Освітня програма: «Радіоелектронні пристрої, системи та комплекси»
«Радіоелектронні комп'ютеризовані засоби»
(найменування освітньої програми)

Форма навчання: денна

Рівень вищої освіти: перший (бакалаврський)

Харків 2021 рік

Розробники: Душепа В.А., старший викладач каф. 501
(прізвище та ініціали, посада, науковий ступінь та вчене звання)


(підпис)

Робочу програму розглянуто на засіданні кафедри №501 аерокосмічних радіоелектронних систем

(назва кафедри)

Протокол № 12/10-21 від « 26 » серпня 2021 р.

Завідувач кафедри к.т.н., доцент
(науковий ступінь та вчене звання)


(підпис)

С.С. Жила
(ініціали та прізвище)

1. Опис навчальної дисципліни

Найменування показника	Галузь знань, спеціальність, освітня програма, рівень вищої освіти	Характеристика навчальної дисципліни (денна форма навчання)
Кількість кредитів – 3	<p>Галузь знань 17 «Електроніка та телекомунікації»</p> <p>Спеціальність 172 «Телекомунікації та радіотехніка»</p> <p>Освітня програма «Радіоелектронні пристрої, системи та комплекси»,</p> <p>Рівень вищої освіти: перший (бакалаврський)</p>	Цикл професійної підготовки
Кількість модулів – 2		Навчальний рік
Кількість змістовних модулів – 2		2021/2022
Індивідуальне завдання – розрахункова робота		Семестр
Загальна кількість годин – 40 аудиторних занять* / 90 (загальна кількість годин)		1-й
Кількість тижневих годин для денної форми навчання: аудиторних – 2.5 самостійної роботи студента – 3.125		Лекції*
		24 годин
		Практичні, семінарські*
		16 годин
		Лабораторні*
	-	
	Самостійна робота	
	50 годин	
	Вид контролю	
	модульний контроль, іспит	

Співвідношення кількості годин аудиторних занять до самостійної роботи становить: 40 / 50.

*Аудиторне навантаження може бути зменшене або збільшене на одну годину залежно від розкладу занять.

2. Мета та завдання навчальної дисципліни

Мета вивчення: навчити студентів програмувати на мові високого рівня Python.

Завдання: вивчення синтаксису мови Python, структур даних, бібліотеки NumPy, знайомство зі складністю алгоритмів.

Результати навчання:

знати:

- базовий синтаксис мови програмування Python;
- структури даних мови Python.

вміти:

- писати прості програми на Python;
- використовувати середовище програмування Spyder;
- користуватись бібліотеками NumPy та Matplotlib.

мати уявлення:

- про об'єктно-орієнтоване програмування;
- про складність алгоритмів.

Міждисциплінарні зв'язки: немає передумов (опит програмування полегшить проходження курсу, але не є обов'язковим).

3. Програма навчальної дисципліни

Змістовий модуль 1. Синтаксис Python, основні структури даних, налагодження програм.

Вступ. Середовище програмування Spyder. Основи синтаксису Python. Типи даних. Цикли і умовний оператор. Робота зі строками у Python. Функції (у тому числі рекурсивні). Структури даних: list (список), tuple (кортеж), dict (словник) та інші. Змінні (mutable) та незмінні об'єкти. Налгодження (debugging) програм. Стил програмування.

Змістовий модуль 2. Об'єктно-орієнтоване програмування, бібліотека NumPy, складність алгоритмів.

Принципи об'єктно-орієнтоване програмування та його інструменти у Python. Векторні обчислення за допомогою бібліотеки NumPy. Побудова графіків за допомогою Matplotlib. Складність алгоритмів (O-нотація).

4. Структура навчальної дисципліни

Назва змістовного модуля і тем	Кількість годин				
	Усього	У тому числі			
		л	п	лаб.	с. р.
1	2	3	4	5	6
Модуль 1					
Змістовний модуль 1. Синтаксис Python, основні структури даних, налагодження програм.					
		16	8	-	20
Разом за змістовним модулем 1	44	16	8	-	20
Змістовний модуль 2. Об'єктно-орієнтоване програмування, бібліотека NumPy, складність алгоритмів.					
		8	8	-	20
Разом за змістовним модулем 2	36	8	8	-	20
Індивідуальне завдання	10	-	-	-	10
Усього годин	90	24	16	-	50

5. Теми семінарських занять

№ п/п	Назва теми	Кількість годин
1		
2		
	Разом	

6. Теми практичних занять

№ з/п	Назва теми	Кільк. год.
1	2	3
1	Середовище Spyder. Прості розрахунки.	2
2	Цикли.	2
3	Функції (у тому числі рекурсивні).	2
4	Списки (lists), кортежі (tuples).	2
5	Словники (dict). Глобальні змінні.	2
6	Об'єктно-орієнтоване програмування.	2
7	Об'єктно-орієнтоване програмування (продовження). Бібліотека NumPy.	2
8	Бібліотека Matplotlib. Складність алгоритмів.	2
	Разом	16

7. Теми лабораторних занять

№ з/п	Назва теми	Кількість годин
1		-
2		-
	Разом	-

8. Самостійна робота

При виконанні домашніх завдань планується використовувати безкоштовну платформу для автоматичної перевірки коду студентів (наприклад, CodePost.io).

Для кожного домашнього завдання (ДЗ) існує строк задачі (deadline), близько неділі, після якого його не можна здати.

№ з/п	Назва теми	Кількість годин
1	Робота зі строками у Python (ДЗ1)	2
2	Функції (ДЗ2)	8
3	Робота з списками (lists). Рекурсивні функції. (ДЗ3)	5
4	Робота з кортежами (tuples) та словниками (dict). (ДЗ4)	5
5	Об'єктно-орієнтоване програмування (ДЗ5)	10
6	Робота з бібліотекою NumPy. (ДЗ6)	5
7	Побудова графіків за допомогою бібліотеки Matplotlib (ДЗ7)	5
8	Виконання розрахункової роботи (РР) на тему “Розрахунок та відображення статистики по результатах навчання студентів у Excel файлі”.	10
	Разом	50

9. Індивідуальні завдання

Розрахункова робота (РР) на тему “Розрахунок та відображення статистики по результатам навчання студентів у Excel файлі”.

10. Методи навчання

Проведення аудиторних лекцій, практичних занять, виконання домашніх завдань, індивідуальні консультації (при необхідності), самостійна робота студентів за матеріалами, опублікованими кафедрою (методичні посібники).

11. Методи контролю

Проведення поточного контролю, письмового модульного контролю, фінальний контроль у вигляді іспитів.

12. Критерії оцінювання та розподіл балів, які отримують студенти

12.1. Розподіл балів, які отримують студенти (кількісні критерії оцінювання)

Складові навчальної роботи	Бали за одне заняття (завдання)	Кількість занять (завдань)	Сумарна кількість балів
Змістовний модуль 1			
Робота на лекціях	-	-	-
Робота на практиках		4	6
Домашні завдання	4	3	12
Модульний контроль	25	-	25
Змістовний модуль 2			
Робота на лекціях	-	-	-
Робота на практиках		4	6
Домашні завдання	4	4	16
Модульний контроль	25	-	25
Виконання і захист РР	10	1	15
Усього за семестр			60...100*

* Підсумкова оцінка не може бути більшою за 100 балів.

Семестровий контроль (іспит/залік) проводиться у разі відмови студента від балів поточного тестування й за наявності допуску до іспиту/заліку. Під час складання семестрового іспиту/заліку студент має можливість отримати максимум 100 балів.

Білет для іспиту/заліку складається з ...

8 тестів – 40 б

Кілька (скоріше всього, три) завдань на програмування (виконуються за комп'ютером або просто студент пише код на папері) – 60 б

12.2. Якісні критерії оцінювання

Необхідний обсяг знань для одержання позитивної оцінки:

- базовий синтаксис Python;
- типи даних, структури даних Python;
- принципи об'єктно-орієнтовного програмування;
- знання з складності алгоритмів.

Необхідний обсяг вмінь для одержання позитивної оцінки:

- базові навички програмування у Python;
- вміння користуватись бібліотекою NumPy;
- вміння користуватись бібліотекою Matplotlib;
- вміння писати власні найпростіші класи.

12.3 Критерії оцінювання роботи студента протягом семестру

Оцінка “**відмінно**” (“А”, 90-100 балів) виставляється за умов:

- 1). ґрунтовні знання синтаксису Python, типів та структур даних;
- 2). добрі навички програмування, вміння користуватись бібліотеками NumPy та Matplotlib;
- 3). вміння писати власні найпростіші класи.

4). вміння читати і розуміти чужий код.

Оцінка “добре” (“В”, 83-89 балів) виставляється за умов:

- 1). доволі ґрунтовні знання синтаксису Python, типів та структур даних;
- 2). добрі навички програмування;
- 3). розуміння принципів об’єктно-орієнтовного програмування.
- 4). вміння читати і розуміти чужий код.

Оцінка “добре” (“С”, 75-82 балів) виставляється за умов:

- 1). задовільні знання синтаксису Python, типів та структур даних;
- 2). можливість писати прості програми на Python;
- 3). розуміння принципів об’єктно-орієнтовного програмування.
- 4). вміння читати і розуміти чужий код.

Оцінка “задовільно” (“D”, 68-74 балів) виставляється за умов:

- 1). мінімально задовільні знання синтаксису Python, типів та структур даних;
- 2). можливість писати найпростіші програми на Python;
- 3). вміння читати і розуміти чужий код.

Шкала оцінювання: бальна і традиційна

Сума балів	Оцінка за традиційною шкалою	
	Іспит, диференційований залік	Залік
90 – 100	Відмінно	Зараховано
75 – 89	Добре	
60 – 74	Задовільно	
0 – 59	Незадовільно	Не зараховано

13. Методичне забезпечення

Усі матеріали викладач послідовно викладає для студентів у системі Ментор за посиланням <https://mentor.khai.edu/course/view.php?id=5824> включаючи

- презентації до усіх лекцій (також доступні записи відео з минулого року);
- презентації до усіх практик;
- 7 домашніх завдань та рішення (видаються викладачем при закінченні строку здачі домашнього завдання).

14. Рекомендована література

Базова

1. Сторінка курсу у системі Mentor (Moodle), НАУ “ХАІ”, Харків, 2020 осінь, Душепа В.А. (<https://mentor.khai.edu/course/view.php?id=5824>)
2. Guttag J.V. Introduction to Computation and Programming Using Python: With Application to Understanding Data, Second Edition. — The MIT Press, 2016. — 591 р.

Допоміжна

2. Матеріали “Introduction to Computer Science and Programming in Python” курсу MIT, 2016 Fall, MIT OpenCourseWare (<https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-0001-introduction-to-computer-science-and-programming-in-python-fall-2016/>)

15. Інформаційні ресурси

- 1). <https://mentor.khai.edu/course/view.php?id=5824>