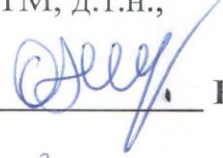


**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ АВТОМОБІЛЬНО-
ДОРОЖНИЙ УНІВЕРСИТЕТ**

Кафедра комп'ютерних технологій і мехатроніки

«ЗАТВЕРДЖУЮ»

Гарант освітньо-професійної
програми «Програмне забезпечення
систем» першого (бакалаврського)
рівня вищої освіти, завідувач
кафедри КТМ, д.т.н.,

професор  Ніконов
О.Я.

« 9 » червня 2020р.

**СИЛАБУС
ПРОГРАМУВАННЯ НА МОВІ PYTHON
PYTHON PROGRAMMING
SYLLABUS**

освітній ступінь	бакалавр / bachelor
галузь знань	12 Інформаційні технології / Information Technology
спеціальність	121 Інженерія програмного забезпечення / Software Engineering
освітня програма	Програмне забезпечення систем / Systems Software

Харків 2020рік

Автори: Пронін С.В. доцент асистент кафедри комп'ютерних технологій і мехатроніки

Силабус розглянуто та затверджено на засіданні кафедри комп'ютерних технологій та мехатроніки 9.06.2020 р. протокол №18

СИЛАБУС

ПРОЕКТУВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

PYTHON PROGRAMMING

SYLLABUS

освітній ступінь	бакалавр / bachelor
галузь знань	12 Інформаційні технології / Information Technology
спеціальність	121 Інженерія програмного забезпечення / Software Engineering
освітня програма	Програмне забезпечення систем / Systems Software

Анотація курсу

1. Викладачі:

Лектор Пронін Сергій Вікторович,

- Доцент кафедри комп'ютерних технологій та мехатроніки;
- Педагогічний стаж – 16 років;
- Контактний телефон : 057-707-37-43;
- e-mail: sergiy9977@ukr.net;
- наукові інтереси: машинне навчання, штучний інтелект;
- стажування та підвищення кваліфікації.

2. Дисципліна: Проектування та конструювання програмного забезпечення

- рік навчання 4
- семестр навчання 7
- кількість кредитів 4
- кількість годин за семестр:
 - Лекційних 16
 - Лабораторних 32
 - На самостійне опрацювання 42
- кількість аудиторних годин на тиждень
 - лекційних 1
 - лабораторних 2

3. Час та місце проведення:

Аудиторні заняття – відповідно до розкладу ХНАДУ з урахуванням специфіки дисципліни проведення дисципліни передбачено в аудиторіях 123, 214, 313;

Позааудиторна робота – самостійна робота студента

4. Пререквизити та постреквизити навчальної дисципліни:

- **Пререквизити:** «Алгоритмізація та програмування», «Об'єктно-орієнтоване програмування», «Проектування програмного забезпечення»
- **Постреквизити:** «Методи та системи штучного інтелекту», «Дипломна робота».

5. Характеристика дисципліни:

5.1 Призначення навчальної дисципліни: Навчальна дисципліна «Програмування на мові Python» є вибірковою дисципліною і входить в професійний цикл, який формує базовий рівень знань для освоєння загально-професійних навичок.

5.2. Мета вивчення навчальної дисципліни «Програмування на мові Python» полягає в вивченні та освоєнні мови програмування Python. є формування в студентів загальнокультурних і професійних компетенцій,

знань, умінь і навичок володіння мовою програмування Python та ефективного її використання для розв'язку прикладних завдань.

5.3 Задачі вивчення дисципліни: По завершенні вивчення дисципліни студенти повинні:

- уміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення
- вміти розробляти людино-машинний інтерфейс;
- знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань;
- мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення
- вміти застосовувати методи компонентної розробки програмного забезпечення;
- знати та вміти застосовувати методи верифікації та валідації програмного забезпечення.

5.4 Зміст навчальної дисципліни: відповідає навчальній та робочій програмі, яка відповідає запитам роботодавців

5.5 План вивчення дисципліни

Результати навчання	Навчальна діяльність	Робочий час студента (год.)	Оцінювання (бал)
Загальні та спеціальні компетентності: здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування Результати навчання: мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення	Тема 1. Введення в мову програмування Python <i>План лекції:</i> - мова програмування Python; - основні структури даних у Python; - інтерпретатор Python; - сфери застосування мови Python Список рекомендованих джерел: <i>Основний: 1,2,3</i> <i>Додатковий: 1,2</i> <i>Інтернет-ресурси: 1</i>	12	6
	Практичне заняття 1. Синтаксис мови Python <i>Мета роботи:</i> Познайомитися із середовищем розробки Python. <i>Завдання:</i> Вивчити основні типи даних, команди введення і виведення даних		

.	Завдання для самостійної роботи. Самостійне опрацювання літературних джерел, які зазначені у списку <i>Питання, винесені на самостійне опрацювання:</i> Огляд стандартної бібліотеки Python		
Загальні та спеціальні компетентності: здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення Результати навчання: уміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення	Тема 2. Математичні та логічні операції в Python <i>План лекції:</i> - уніфікована мова моделювання; - статичні моделі об'єктно-орієнтованих програмних систем; - динамічні моделі об'єктно-орієнтованих програмних систем <i>Список рекомендованих джерел:</i> <i>Основний:</i> 2,3 <i>Додатковий:</i> 1,2 <i>Інтернет-ресурси:</i> 1	14	12
	Практичне заняття 2. Математичні операції в Python <i>Мета роботи:</i> Ознайомитись з основними математичними операторами в Python <i>Завдання:</i> Написати програму, яка б обчислювала заданий арифметичне вираз при заданих змінних		
	Завдання для самостійної роботи. Самостійне опрацювання літературних джерел, які зазначені у списку <i>Питання, винесені на самостійне опрацювання:</i> Огляд бібліотеки math		
	Практичне заняття 3. Логічні операції в Python <i>Мета роботи:</i> Ознайомитись з основними логічними операторами в Python <i>Завдання:</i> Навчитись застосовувати логічні оператори		

	Завдання для самостійної роботи. Самостійне опрацювання літературних джерел, які зазначені у списку <i>Питання, винесені на самостійне опрацювання:</i> Огляд бібліотеки math		
Загальні та спеціальні компетентності: здатність розробляти архітектури, модулі та компоненти програмних систем Результати навчання: знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань	Тема 3. Операції з потоками виведення в Python <i>План лекції:</i> - потоки виведення в Python; - розгалуження в Python; - цикли в мові Python; Список рекомендованих джерел: <i>Основний:</i> 2,3 <i>Додатковий:</i> 2 <i>Інтернет-ресурси:</i>	14	12
	Практичне заняття 4. Структура розгалуження в Python <i>Мета роботи:</i> Познайомитися зі структурою розгалуження (if, if-else, if-elif-else) <i>Завдання:</i> Вирішити індивідуальні завдання використовуючи розгалуження		
	Завдання для самостійної роботи. Самостійне опрацювання літературних джерел, які зазначені у списку <i>Питання, винесені на самостійне опрацювання:</i> Рішення задач за допомогою розгалуження		
	Практичне заняття 5. Цикли в мові Python <i>Мета роботи:</i> Отримати практичні навички щодо застосування циклів в мові Python <i>Завдання:</i> Вирішити індивідуальні завдання використовуючи цикли		

	Завдання для самостійної роботи. Самостійне опрацювання літературних джерел, які зазначені у списку <i>Питання, винесені на самостійне опрацювання:</i> Рішення задач за допомогою циклів		
Загальні та спеціальні компетентності: здатність розробляти архітектури, модулі та компоненти програмних систем Результати навчання: знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань	Тема 4. Структури даних. Рядки та списки <i>План лекції:</i> - структури даних; - рядки; - списки; Список рекомендованих джерел: <i>Основний: 1,2,3</i> <i>Додатковий: 1,2</i> <i>Інтернет-ресурси: 1</i>	18	15
	Практичне заняття 6. Робота з рядками в Python <i>Мета роботи:</i> Познайомиться з методами роботи з рядками в мові програмування Python <i>Завдання:</i> Вирішити індивідуальні завдання використовуючи рядки		
	Завдання для самостійної роботи. Самостійне опрацювання літературних джерел, які зазначені у списку <i>Питання, винесені на самостійне опрацювання:</i> Вбудовані типи даних		
	Практичне заняття 7. Операції над списками в Python <i>Мета роботи:</i> Вивчення списків в Python <i>Завдання:</i> Вирішити індивідуальні завдання використовуючи списки		
	Завдання для самостійної роботи. Самостійне опрацювання літературних джерел, які зазначені у списку <i>Питання, винесені на самостійне опрацювання:</i> Використання списків		

Загальні та спеціальні компетентності: здатність розробляти архітектури, модулі та компоненти програмних систем Результати навчання: знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань	Тема 5. Структури даних. Кортежі. Словники <i>План лекції:</i> - структури даних; - кортежі; - словники; - безлічі Список рекомендованих джерел: <i>Основний: 1, 2, 3</i> <i>Додатковий: 1</i> <i>Інтернет-ресурси: 1</i>	12	10
	Практичне заняття 8. Структури даних. Безлічі, кортежі, словники в Python <i>Мета роботи:</i> Отримати практичні навички застосування безлічів, кортежів, словників в Python <i>Завдання:</i> Вирішити індивідуальні завдання використовуючи безлічі, кортежі та словники		
	Завдання для самостійної роботи. Самостійне опрацювання літературних джерел, які зазначені у списку <i>Питання, винесені на самостійне опрацювання:</i> Огляд операцій для роботи зі словниками		
Загальні та спеціальні компетентності: здатність розробляти архітектури, модулі та компоненти програмних систем; здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його	Тема 6. Модулі в Python <i>План лекції:</i> - функції в Python; - модулі в Python; Список рекомендованих джерел: <i>Основний: 1,3</i> <i>Додатковий: 1</i> <i>Інтернет-ресурси: 1</i>	18	15

структури, поведінки та процесів функціонування Результати навчання: знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань; вміти застосовувати методи компонентної розробки програмного забезпечення	Практичне заняття 9. Функції та модулі в Python Мета роботи: Отримати практичний досвід зі створення функцій і модулів в Python Завдання: Написати програму використовуючи функції і модулі відповідно до індивідуального завдання		
	Завдання для самостійної роботи. Самостійне опрацювання літературних джерел, які зазначені у списку <i>Питання, винесені на самостійне опрацювання:</i> Модулі стандартної бібліотеки		
	Практичне заняття 10. Робота з файлами в Python Мета роботи: Отримання досвіду роботи з файлами в Python Завдання: Написати програму по обробці інформації з файлів		
	Завдання для самостійної роботи. Самостійне опрацювання літературних джерел, які зазначені у списку <i>Питання, винесені на самостійне опрацювання:</i> Функції вводу-виводу		
Загальні та спеціальні компетентності: здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування;	Тема 7. Об'єктно-орієнтоване програмування на Python <i>План лекції:</i> - введення в об'єктно-орієнтоване програмування на Python; - інкапсуляція; - поліморфізм; - наслідування Список рекомендованих джерел: <i>Основний: 1, 3</i> <i>Додатковий: 1,2</i> <i>Інтернет-ресурси 1</i>	20	20

здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення; здатність розробляти архітектури, модулі та компоненти програмних систем	Практичне заняття 11. Об'єктно-орієнтоване програмування в мові Python Мета роботи: Отримати практичний досвід створення класів у мові Python. Завдання: Написати програму за допомогою класів		
Результати навчання: уміння вибирати та використовувати відповідну задачу методологію створення програмного забезпечення; знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань; вміти застосовувати методи компонентної розробки програмного забезпечення; знати та вміти застосовувати методи верифікації та валідації програмного забезпечення	Завдання для самостійної роботи. Самостійне опрацювання літературних джерел, які зазначені у списку <i>Питання, винесені на самостійне опрацювання:</i> Об'єктно-орієнтоване програмування в мові Python		
Загальні та спеціальні компетентності: здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення Результати навчання: вміти розробляти людино-машинний інтерфейс	Тема 8. Створення графічного інтерфейсу <i>План лекції:</i> - введення у бібліотеку Tkinter; - віджети; - зміна параметрів за замовчуванням; - методи в Tkinter Список рекомендованих джерел: <i>Основний: 2</i> <i>Додатковий: 1</i> <i>Інтернет-ресурси 1</i>	12	10

	Практичне заняття 12. Розробка додатків з графічним інтерфейсом на мові Python <i>Мета роботи:</i> Отримати практичний навик по створенню графічного інтерфейсу на мові Python. <i>Завдання:</i> Розробити графічний інтерфейс програми		
	Завдання для самостійної роботи. Самостійне опрацювання літературних джерел, які зазначені у списку <i>Питання, винесені на самостійне опрацювання:</i> Огляд патерну MVC на мові Python		
Разом		120/4	100
Підсумковий контроль		Письмовий екзамен	

6. Рекомендовані джерела інформації

Основна

1. Swaroop Chitlur A Byte of Python/ Swaroop Chitlur (Перевод: Владимир Смоляр) – Sphinx -2020-164 с.
2. Федоров, Д. Ю. Программирование на языке высокого уровня Python : учебное пособие для прикладного бакалавриата / Д. Ю. Федоров. — М. : Издательство Юрайт, 2018. — 126 с.
3. Буйначев, С. К. Основы программирования на языке Python : учебное пособие / С. К. Буйначев, Н. Ю. Боклаг. – Екатеринбург : Изд-во Урал. ун-та, 2014. – 91

Допоміжна

1. Python documentation URL: <https://docs.python.org/3.7/library/functions.html> (Дата звернення: 16.12.2020г.).
2. Марк Лутц Изучаем Python, 3е издание/ Марк Лутц(Перевод А. Киселева) - СПб.: СимволПлюс, 2009. – 848 с.

7. Контроль та оцінювання результатів навчання:

Під час вивчення дисципліни «Проектування та конструювання програмного забезпечення» викладачем здійснюється поточний та підсумковий контроль, що передбачає:

- перевірку рівня засвоєння теоретичного матеріалу (усне та письмове опитування);
- захист лабораторних робіт (протягом семестру)

8. Критерії оцінювання результатів навчання Підсумковий контроль знань та компетентностей студентів з навчальної дисципліни здійснюється на підставі іспиту.

Відповідність підсумкової семестрової рейтингової оцінки в балах оцінці за національною шкалою та шкалою ECTS:

Оцінка в балах	Оцінка за національною шкалою	Оцінка за шкалою ECTS	
		Оцінка	Пояснення
90-100	Відмінно	A	Відмінно (відмінне виконання лише з незначною кількістю помилок)
82 – 89	Добре	B	Дуже добре (вище середнього рівня з кількома помилками)
75 – 81		C	Добре (в загальному вірне виконання з певною кількістю суттєвих помилок)
67 – 74	Задовільно	D	Задовільно (непогано, але зі значною кількістю недоліків)
60 – 66		E	Достатньо (виконання задовольняє мінімальним критеріям)
35 – 59	Незадовільно	FX	Незадовільно (з можливістю повторного складання)
1 – 34		F	Незадовільно (з обов'язковим повторним курсом)