

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА

Факультет радіофізики, електроніки та комп'ютерних систем

Кафедра комп'ютерної інженерії

«ЗАТВЕРДЖУЮ»

Заступник декана з навчальної роботи

_____ Наталія ГОРБОВЦОВА

« ____ » _____ 2023 року

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Програмування

для студентів

галузь знань	12 “Інформаційні технології”
спеціальність	123 “Комп'ютерна інженерія”
рівень вищої освіти	перший (бакалаврський) рівень
освітня програма	“Інженерія комп'ютерних систем і мереж”
вид дисципліни	обов'язкова

Форма навчання	денна
Навчальний рік	2023/2024
Семестр	1
Кількість кредитів ECTS	4
Мова викладання	українська
Форма заключного контролю	залік

Викладач:

Юрій ЮРЧИК, асистент кафедри комп'ютерної інженерії

Пролонговано: на 20__/20__ н. р. _____ (_____) « ____ » _____ 20__ р.

на 20__/20__ н. р. _____ (_____) « ____ » _____ 20__ р.

Розробник:

Юрій ЮРЧИК, асистент кафедри комп'ютерної інженерії

«ЗАТВЕРДЖЕНО»

Завідувач кафедри комп'ютерної інженерії

_____ **Юрій БОЙКО**

Протокол № __ від « ____ » _____ 2023 р.

Схвалено науково-методичною комісією факультету радіофізики, електроніки та комп'ютерних систем

Протокол № __ від « ____ » _____ 2023 р.

Голова науково-методичної комісії

Сергій РАДЧЕНКО

« ____ » _____ 2023 р.

ВСТУП

1. Мета дисципліни – ознайомлення студентів із основними принципами створення програмного забезпечення.

2. Попередні вимоги до опанування або вибору навчальної дисципліни:

Навчальна дисципліна «Програмування» передбачає наявність у слухачів базових навичок роботи з персональною електронною обчислювальною технікою (ЕОМ).

Попередні вимоги:

студент повинен вміти: використовувати ЕОМ на рівні користувача.

3. Анотація навчальної дисципліни:

Дисципліна «Програмування» знайомить студентів із основними засадами програмування. Вивчення дисципліни передбачає отримання як теоретичних, так і практичних навичок у створенні програмного забезпечення (ПЗ).

У рамках теоретичних засад, лекційний матеріал передбачає надання знань про основні завдання та цілі програмування як такого. Розглядається типізація мова програмування, їх особливості. Обґрунтовується необхідність використання базових інструментів (компілятор, дебагер, тощо) та їх еволюційний розвиток.

Переважає більшість проектів, пов'язаних з розробкою ПЗ, передбачає колективну роботу над однією задачею групи спеціалістів. Необхідною умовою для цього є використання систем контролю версій ПЗ та колективної взаємодії. У дисципліні розгадається засіб Git, принципи його роботи та особливості застосування.

Основою практичної частини дисципліни є виконання студентами лабораторних робіт, завдання яких передбачає створення ПЗ згідно заданих критеріїв. У якості інструменту пропонується використання мови програмування Python.

Дисципліною розглядаються базові засади мови програмування Python. Лекційний матеріал передбачає розгляд синтаксису мови, її виразів, дослідження стандартних та сторонніх програмних бібліотек.

4. Завдання навчальної дисципліни (навчальні цілі):

1. Надати основні теоретичні відомості про принципи створення ПЗ, його структуру та базовий інструментарій.

2. Розглянути сучасний стан галузі програмування, тенденції її розвитку. Надати типізацію сучасних мов програмування, їх базові характеристики.

3. Надати базові знання про використання інструментів контролю версій ПЗ (Git) та колективної взаємодії (GitHub, GitLab).

4. Надати базові знання про мову програмування Python. Розгадаються її синтаксис, основні вирази та особливості, досліджуються стандартна та сторонні програмні бібліотеки.

Дисципліна спрямована на формування програмних компетентностей:

ЗК1. Здатність до абстрактного мислення, аналізу і синтезу

ЗК2. Здатність вчитися і оволодівати сучасними знаннями.

ЗК3. Здатність застосовувати знання у практичних ситуаціях.

ФК2. Здатність використовувати сучасні методи і мови програмування для розроблення алгоритмічного та програмного забезпечення.

ФК3. Здатність створювати системне та прикладне програмне забезпечення комп'ютерних систем та мереж.

ФК11. Здатність оформляти отримані робочі результати у вигляді презентацій, науково-технічних звітів.

5. Результати навчання за дисципліною:

Результат навчання (1, знати; 2, вміти; 3, комунікація; 4, автономність та відповідальність)		Форми (та/або методи і технології) викладання і навчання	Методи оцінювання та пороговий критерій оцінювання (за необхідності)	Відсоток у підсумковій оцінці з дисципліни
Код	Результат навчання			
1	знати:	лекційні заняття	залік	до 30
1.1	Принципи побудови програмного забезпечення. Типізації мов програмування.	лекція	залік	5
1.2	Роботу засобів компіляції, особливості їх роботи.	лекція	залік	4
1.3	Принципи роботи систем контролю версій програмного забезпечення та засобів колективної взаємодії розробників.	лекція	залік	5
1.4	Базові вирази та синтаксис мови програмування Python.	лекція	залік	3
1.5	Структура та можливості бібліотеки мови Python, особливості та область їх застосування.	лекція	залік	5
1.6	Підходи до синтезу алгоритмів. Патерни вирішення типових задач.	лекція	залік	4
1.7	Тенденції розвитку методів та засобів створення програмного забезпечення.	лекція	залік	4
2	вміти:	лекційні демонстрації та лабораторні роботи	захист звітів лабораторних робіт	до 60
2.1	Обирати програмні засоби для вирішення поставленого технічного завдання, обґрунтувати обране рішення.	лабораторні роботи	звіти з лабораторних робіт	10
2.2	Використовувати засоби контролю версії програмного забезпечення та колективної взаємодії розробника.	лекційні демонстрації та лабораторні роботи	звіти з лабораторних робіт	10
2.3	Використовувати інтегровані середовища розробки для вирішення задач створення програмного забезпечення.	лекційні демонстрації та лабораторні роботи	звіти з лабораторних робіт	10
2.4	Створювати алгоритм та його програмну реалізацію на прикладі мови програмування Python.	лабораторні роботи	звіти з лабораторних робіт	10
2.5	Використовувати стандартну та сторонні програмні бібліотеки мови програмування Python для вирішення поставлених задач.	лабораторні роботи	звіти з лабораторних робіт	10
2.6	Проводити аналіз програмного забезпечення на наявність помилок, їх виправлення.	лабораторні роботи	звіти з лабораторних робіт	10
3	комунікація:	лекційні заняття та лабораторні роботи	захист звітів лабораторних робіт	до 5
3.1	Здатність будувати грамотну науково-технічну комунікацію в усній та письмовій формах, підбирати правильну термінологію	лекція	звіти з лабораторних робіт	3
3.2	Здатність до командної роботи у промислових IT-проектах	лекція	звіти з лабораторних робіт	2
4	автономність та відповідальність:	лекційні демонстрації та лабораторні роботи	захист звітів лабораторних робіт	до 5
4.1	Здатність до самостійного пошуку технічної літератури або інших джерел інформації для пошуку шляхів усунення помилок студентом під час виконання лабораторних робіт	лекційні демонстрації та лабораторні роботи	звіти з лабораторних робіт	5

6. Співвідношення результатів навчання дисципліни із програмними результатами навчання

Результати навчання дисципліни	Код															
	1.1	1.2	1.3	1.4	1.5	1.6	1.7	2.1	2.2	2.3	2.4	2.5	2.6	3.1	3.2	4.1
Програмні результати навчання (назва)																
ПРН3. Знати новітні технології в галузі комп'ютерної інженерії.	+	+	+				+	+	+							
ПРН6. Вміти застосовувати знання для ідентифікації, формулювання і розв'язування технічних задач спеціальності, використовуючи методи, що є найбільш придатними для досягнення поставлених цілей.	+	+	+				+	+			+	+	+			
ПРН10. Вміти розробляти програмне забезпечення для вбудованих і розподілених застосувань, мобільних і гібридних систем, розраховувати, експлуатувати типове для спеціальності обладнання.				+	+	+				+	+	+	+			
ПРН11. Вміти здійснювати пошук інформації в різних джерелах для розв'язання задач комп'ютерної інженерії.																+
ПРН14. Вміти поєднувати теорію і практику, а також приймати рішення та виробляти стратегію діяльності для вирішення завдань спеціальності з урахуванням загальнолюдських цінностей, суспільних, державних та виробничих інтересів.															+	
ПРН18. Використовувати інформаційні технології та для ефективного спілкування на професійному та соціальному рівнях.														+		
ПРН19. Здатність адаптуватись до нових ситуацій, обґрунтовувати, приймати та реалізовувати у межах компетенції рішення.																+

7. Схема формування оцінки

7.1. Форми оцінювання

Рівень досягнення всіх запланованих результатів навчання визначається за результатами виконання та захисту звітів лабораторних робіт, а також за результатами підсумкового заліку. Внесок результатів навчання у підсумкову оцінку, за умови їх опанування на належному рівні:

- результати навчання 1.1 – 1.7 [знання] – до 30%;
- результати навчання 2.1 – 2.6 [вміння] – до 60%;
- результати навчання 3.1 – 3.2 [комунікація] – до 5%;
- результат навчання 4.1 [автономність та відповідальність] – до 5%;

Форми оцінювання:

- **семестрове оцінювання:** Навчальний семестр має 6 лабораторних робіт та дві модульні контрольні роботи. Після завершення виконання лабораторних робіт та написання звітів проводяться усні захисти звітів із демонстраціями коректного виконання завдань. Обов'язковим для допуску до заліку є: захист звіту з кожної лабораторної роботи з кількістю балів не менше 60% від максимально можливої оцінки.
- **підсумкове оцінювання (у формі заліку):** форма заліку – письмове електронне тестування. Тест складається з 30 питань із варіантами відповіді, які оцінюються по 1 балу за кожне зарите питання, та 2 відкритих питань, які оцінюються по 5 балів. Всього за залік можна отримати від 0 до 40 балів. Умовою досягнення позитивної оцінки за дисципліну є отримання не менш ніж 60 балів, оцінка за залік не може бути меншою за **24 бали**.
- **умови допуску до підсумкового заліку:** умовою допуску до заліку є отримання студентом сумарно не менше, ніж *критично-розрахунковий мінімум* за семестр. Студенти, які протягом семестру сумарно набрали меншу кількість балів, ніж критично-розрахунковий мінімум **36 балів** (60% від максимально можливих), для одержання допуску до заліку обов'язково повинні виконати додаткове завдання за тематикою лабораторних робіт.

У випадку відсутності студента з поважних причин відпрацювання та захист пропущених лабораторних робіт здійснюються у відповідності до «Положення про організацію освітнього процесу у Київському національному університеті»

7.2. Організація оцінювання

Оцінювання за формами контролю:

Семестрова робота	Кількість балів	
	Min. – 6	Max. – 10
Лабораторна робота №1	3	5
Лабораторна робота №2	3	5
Лабораторна робота №3	3	5
Лабораторна робота №4	3	5
Лабораторна робота №5	3	5
Лабораторна робота №6	3	5
Модульна контрольна робота 1	9	15
Модульна контрольна робота 2	9	15
Додаткові завдання	–	12

Орієнтований графік оцінювання:

Форма оцінювання	Орієнтовний період для здійснення відповідної форми оцінювання
Лабораторні роботи №1 та №2	вересень
Лабораторні роботи №2 та №3	вересень
Модульна контрольна робота 1	жовтень
Лабораторні роботи №5 та №6	жовтень
Модульна контрольна робота 2	листопад
Добір балів/додаткові завдання	грудень
Залік	грудень

Розрахунок балів, які отримують при успішній здачі заліку:

Значення	Змістовні модулі	Лабораторні роботи	Залік	Підсумкова оцінка
Мінімум	18	18	24	60
Максимум	30	30	40	100

7.3. Шкала відповідності оцінок

Оцінка (за національною шкалою) / National grade	Рівень досягнень, % / Marks, %
Відмінно / Excellent	90-100%
Добре / Good	75-89%
Задовільно / Satisfactory	60-74%
Незадовільно / Fail	0-59%

8. Структура навчальної дисципліни

8.1. Тематичний план занять

№ з/п	Назва теми	Кількість годин		
		Лекції	Лабораторні роботи	Самостійна робота
1	Вступ до курсу. Структура програми. Інструменти розробника.	2	4	6
2	Типи даних та оператори.	2		6
3	Логічні умови та цикли.	2	4	6
4	Функціональне програмування.	2	6	6
5	Об'єктно-орієнтовне програмування.	8	6	8
6	Робота з файловою системою.	2	4	6
7	Стандартні бібліотеки та їх функціонал	2		6
8	Засоби візуалізації результатів обчислень	4		6
9	Реалізація графічного інтерфейсу користувача	2	4	8
10	Створення бібліотек.	2		6
Всього		28	28	64

Загальний обсяг **120** год., в тому числі:
 Лекції **28** год.
 Лабораторні роботи **28** год.
 Самостійна робота **64** год.

8.2. Лабораторні заняття

№ з/п	Назва теми	Кількість годин
1	Базова структура програми.	4
2	Використання умов вибору та циклів у обчисленнях.	4
3	Використання функцій у алгоритмах.	6
4	Парадигма об'єктно-орієнтованого програмування	6
5	Робота з файловою системою.	4
6	Реалізація графічного інтерфейсу користувача	4
Всього		28

9. Рекомендована література:

Основні джерела:

1. Python Software Foundation. Python 3.10.6 documentation [Електронний ресурс] / Python Software Foundation. – 2022. – Режим доступу до ресурсу: <https://docs.python.org/3/>.
2. Kief Morris. Infrastructure as Code, 2nd Edition. — O'Reilly Media Inc., 2020. — 350 pp.
3. Gigi Sayfan. Mastering Kubernetes – Third Edition: Level Up Your Container Orchestration Skills with Kubernetes to Build, Run, Secure, and Observe Large-scale Distributed Apps. — Packt Publishing, 2020. — 642 pp.
4. Thomas Uphill, John Arundel, Neependra Khare. DevOps: Puppet, Docker, and Kubernetes. — Packt Publishing, 2017. — 924 pp.

Додаткові і джерела:

1. Shashank Mohan Jain. Linux Containers and Virtualization: A Kernel Perspective. — Apress, 2020. — 148 pp.
2. Gene Kim, Jez Humble, Patrick Debois, John Willis. The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations. — IT Revolution Press, 2016. — 480 pp.
3. Mikolaj Pawlikowski. Chaos Engineering: Site reliability through controlled disruption. — Simon and Schuster, 2021. — 424 pp.