

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЗАХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ВСП «ФАХОВИЙ КОЛЕДЖ ЕКОНОМІКИ, ПРАВА ТА  
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ЗУНУ»



**ЗАТВЕРДЖУЮ**

**Директор ВСП «ФКЕПІТ ЗУНУ»**

**Василь МАРТИНЮК**

**«31» серпня 2023р.**

**РОБОЧА ПРОГРАМА  
З ДИСЦИПЛІНИ  
«Мережеве програмування»**

Освітньо-професійний ступінь – фаховий молодший бакалавр

Галузь знань: *12 Інформаційні технології*

Спеціальність: *123 Комп'ютерна інженерія*

Освітньо-професійна програма: *«Комп'ютерна інженерія»*

**Циклова комісія комп'ютерної інженерії**

Форма навчання	Курс	Семестр	Лекції (год.)	Практичні (год.)	Самостійна робота	Загальний обсяг, годин	Екзамен/ залік
Денна	4	8	22	33	65	120	Залік

**Тернопіль – 2023 р.**

Робоча програма складена на основі освітньо-професійної програми підготовки фахового молодшого бакалавра галузі знань **12 Інформаційні технології**, спеціальності **123 Комп'ютерна інженерія**, затвердженої Педагогічною радою ВСП «ФКЕПІТ ЗУНУ» «26» серпня 2020р.

Робоча програма складена викладачем Мельник. В.О

Робоча програма затверджена на засіданні циклової комісії інформатики та комп'ютерної інженерії, протокол № 1 від 28 серпня 2023р.

Голова циклової комісії



Маркопольський С.В.

**СТРУКТУРА РОБОЧОЇ ПРОГРАМИ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ  
«Мережеве програмування»**

**1. ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

<b>Найменування показників</b>	<b>Галузь знань, спеціальність, освітньо-професійна програма, освітньо-професійний ступінь</b>	<b>Характеристика навчальної дисципліни</b>
		<i>Денна форма навчання</i>
Змістових модулів - 3	Галузь знань: <b>12 Інформаційні технології</b>	<b>Нормативна</b>
Загальна кількість годин - 120	Освітньо-професійна програма «Комп'ютерна інженерія» Освітньо-професійний ступінь: фаховий молодший бакалавр	<b>Лекції (22 год.)</b>
		<b>Практичні (33 год.)</b>
		<b>Самостійна робота (65 год.)</b>
		<b>Вид контролю: залік</b>

## 2. МЕТА ТА ЗАВДАННЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

**Мета:** підготовка бакалаврів до успішної роботи використання практичних навичок на основі формування у студентів системи знань, використання теорії та навичок методів проектування програмних додатків на основі об'єктно-орієнтованого підходу .

Метою дисципліни є поглиблене вивчення сучасних технологій створення мережевих програмних продуктів на основі об'єктно-орієнтованого підходу.

**Завдання:** вивчення дисципліни полягає в ознайомленні студентів з основами побудови та проектування мережевих додатків для комп'ютерних систем, а також прищеплення практичних навичок роботи з існуючими мережевими додатками та системами.

У результаті вивчення навчальної дисципліни «Мережеве програмування» студент повинен:

**знати:**

- принципи організації і функціонування локальних і глобальних мереж.
- принципи проектування програмного забезпечення в сучасній індустрії інформаційних технологій.
- сучасні термінології і методи автоматизації розробки програмного забезпечення.
- вивчення сучасних моделей доступу до даних в глобальних інформаційних мережах на прикладі мережі Інтернет.

**вміти:**

- працювати з конкретними технологіями мережевого програмування
- аналізувати узагальнену структуру телекомунікаційних технологій та її технічні показники
- Аналізувати проблеми щодо створення програмного забезпечення інформаційних систем.
- Використовувати інформаційні та комунікативні технології при обміні та обробці інформації.

**Найменування та опис компетентностей, формування котрих забезпечує вивчення дисципліни «Мережеве програмування »**

- СК2. Здатність застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування апаратних, програмних та інструментальних засобів комп'ютерної інженерії.
- СК16. Здатність здійснювати моделювання процесів і об'єктів з використанням передових здобутків програмних та апаратних технологій.

**Результати навчання:**

- РН2. Знати і розуміти теоретичні положення, що лежать в основі функціонування апаратних та програмних засобів комп'ютерної інженерії.
- РН6. Тестувати, діагностувати та обслуговувати апаратні та програмні засоби комп'ютерної інженерії.
- РН7. Застосувати знання для формулювання і розв'язування технічних задач спеціальності, використовуючи методи, що є найбільш придатними для досягнення

поставлених цілей.

- РН9. Розробляти, тестувати, впроваджувати, експлуатувати програмне забезпечення для вбудованих та розподілених систем.

- РН11. Ідентифікувати, класифікувати та описувати роботу програмно-технічних засобів комп'ютерної інженерії.

- РН12. Поєднувати теорію і практику, знаходити та обґрунтовувати шляхи рішення типових задач у професійній діяльності з урахуванням виробничих інтересів.

- РН14. Використовувати сучасні інтегровані середовища, методи і технології розробки, впровадження, адміністрування комп'ютерних систем та мереж, без даних і знань.

- РН16. Спілкуватись усно та письмово з професійних питань українською та іноземними мовами.

- РН17. Вміти обґрунтовувати прийняті рішення, оцінювати, оформляти та представляти результати професійної діяльності згідно діючій нормативній документації.

- РН18. Вміти використовувати сучасні інтегровані середовища, методи і технології розробки, впровадження, адміністрування комп'ютерних систем та мереж, баз даних і знань.

### **3. ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ «Мережеве програмування»**

**Змістовний модуль 1. Концептуальні основи системного програмного забезпечення.**

#### **Тема 1. Вступ в Java.**

Історія виникнення. Основні характеристики мови. Відмінності Java від C++. Структура Java програми. Лексичні основи.

#### **Тема 2. Типи даних, змінні і літерали.**

Прості типи. Приведення типів. Арифметичні оператори. Математичні функції. Бітові оператори. Логічні оператори. Оператори управління. Циклічні оператори.

#### **Тема 3. Алгоритми роботи з матрицями та стрічками в Java.**

Створення матриць. Заповнення матриць. Пошук в матриці. Сортування матриці. Динамічні масиви. Оголошення масиву типу string. Операції над символьними рядками. Клас StringBuffer.

#### **Тема 4. Класи та об'єкти.**

Оголошення класу та об'єктів. Методи класу. Оголошення конструктора.

#### **Тема 5. Наслідування в класах.**

Наслідування. Перевантаження та переозначення методів. Фіналізація об'єкта. Абстрактні класи.

**Змістовний модуль 2. Керування ресурсами та пам'яттю операційних систем.**

**Тема 6. Пакети та інтерфейси.**

Пакети. Інтерфейси.

**Тема 7. Лямбда-вирази**

Поняття лямбда-виразу. Структура лямбда-виразу. Передача параметрів лямбда-виразу. Блоки коду в межах лямбда виразу.

**Тема 8. Обробка виключень.**

Основи теорії виключень. Оператори підсистеми обробки КС.

**Тема 9. Механізм колекцій.**

Поняття колекції. Типи колекції. Інтерфейс Collection. Переваги та недоліки застосування.

**Тема 10. Файли.**

Файли та каталоги. Класи OutputStream та InputStream. Файлові потоки. Фільтровані потоки.

**Змістовний модуль 3. Архітектури операційних систем.**

**Тема 11. Візуальні компоненти графічного дизайну користувача.**

Поняття графічного інтерфейсу. Пакет java.awt. Пакет java.swing. Обробка повідомлень.

**Тема 12. Кросплатформенні додатки з JavaFX.**

Поняття JavaFX. Мова розмітки FXML. Панелі компоновки. Елементи керування.

**Тема 13. Мережеве програмування засобами Java.**

Поняття комп'ютерної мережі. Пакет java.net. Сокети.

**Тема 14. Потоки.**

Поняття потоку. Створення потоків. Керування потоками.

#### 4. СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

№	Назви змістових модулів і тем	КІЛЬКІСТЬ ГОДИН		
		Лекції	Практичні (лабораторні) заняття	Самостійна робота
1	2	3	4	5
<b>Змістовний модуль 1. Концептуальні основи системного програмного забезпечення.</b>				
1.	Тема 1. Вступ в Java.	1	2	5
2.	Тема 2. Типи даних, змінні і літерали.	1	2	5
3.	Тема 3. Алгоритми роботи з матрицями та стрічками в Java.	1	2	5
4.	Тема 4. Класи та об'єкти.	1	2	5
5.	Тема 5. Наслідування в класах.	2	2	5
<b>Змістовний модуль 2. Керування ресурсами та пам'яттю операційних систем.</b>				
6.	Тема 6. Пакети та інтерфейси.	2	2	5
7.	Тема 7. Лямбда-вирази.	2	3	5
8.	Тема 8. Обробка виключень.	2	2	5
9.	Тема 9. Механізм колекцій.	2	2	5
10.	Тема 10. Файли.	2	2	5
<b>Змістовний модуль 3. Архітектура операційних систем.</b>				
11.	Тема 11. Візуальні компоненти графічного	2	3	5

	інтерфейсу користувача.			
12.	Тема 12. Кросплатформенні додатки з JavaFX .	2	3	5
13.	Тема 13. Мережеве програмування засобами Java.	1	3	3
14.	Тема 14. Потоки.	1	3	2
Разом		22	33	65

## **5. ТЕМИ ПРАКТИЧНИХ ЗАНЯТЬ з дисципліни «Мережеве програмування»**

### **Практичне заняття 1**

**Тема 1. Ознайомлення з середовищем розробки програмного забезпечення (ПЗ) IntelliJIDEA.**

#### **План практичного заняття**

1. Середовище розробки IntelliJIDEA.
2. Структура програми на Java.
3. Типи даних мови Java.

### **Практичне заняття 2**

**Тема 2. Одномірні та багатомірні масиви. Робота зі стрічками.**

#### **План практичного заняття**

1. Особливості реалізації масивів на Java.
2. Особливості обробки масивів на Java.
3. Особливості роботи з стрічками на Java.

### **Практичне заняття 3**

**Тема 3. Візуальні компоненти на мові програмування Java.**

#### **План практичного заняття**

1. Поняття графічного інтерфейсу користувача.
2. Бібліотека класів awt.
3. Бібліотека класів string.

### **Практичне заняття 4**

**Тема 4. Розробка графічного інтерфейсу з використанням технологій JavaFX.**

#### **План практичного заняття**

1. Особливості технологій JavaFX.
2. Ієрархія класів з технології JavaFX.
3. Особливості застосування класів JavaFX.

### **Практичне заняття 5**



## **Тема 5. Робота з мережею в Java.**

### **План практичного заняття**

1. Поняття «клієнт-серверної» архітектури.
2. Класи для обробки стрічкових даних.
3. Класи для обміну інформацією в мережі.

## **Практичне заняття 6**

### **Тема 6. Багатопоточне програмування в Java.**

#### **План практичного заняття**

1. Поняття потоку.
2. Класи для реалізації розпаралелення програми.
3. Переваги та недоліки розпаралелення.

## **Практичне заняття 7**

### **Тема 7. Лямбда-вирази.**

#### **План практичного заняття**

1. Поняття потоку.
2. Класи для реалізації розпаралелення програми.
3. Переваги та недоліки розпаралелення.

## **Практичне заняття 8**

### **Тема 8. Обробка виключень.**

#### **План практичного заняття**

1. Поняття потоку.
2. Класи для реалізації розпаралелення програми.
3. Переваги та недоліки розпаралелення.

## **Практичне заняття 9**

### **Тема 9. Механізм колекцій.**

#### **План практичного заняття**

1. Поняття потоку.
2. Класи для реалізації розпаралелення програми.
3. Переваги та недоліки розпаралелення.

## **Практичне заняття 10**

### **Тема 10. Файли.**

#### **План практичного заняття**

1. Поняття потоку.
2. Класи для реалізації розпаралелення програми.
3. Переваги та недоліки розпаралелення.

## **Практичне заняття 11**

### **Тема 11. Візуальні компоненти графічного інтерфейсу користувача.**

#### **План практичного заняття**

1. Поняття потоку.
2. Класи для реалізації розпаралелення програми.

3. Переваги та недоліки розпаралелення.

### **Практичне заняття 12**

#### **Тема 12. Кросплатформенні додатки з JavaFX.**

##### **План практичного заняття**

1. Поняття потоку.
2. Класи для реалізації розпаралелення програми.
3. Переваги та недоліки розпаралелення.

### **Практичне заняття 13**

#### **Тема 13. Мережеве програмування засобами Java.**

##### **План практичного заняття**

1. Поняття потоку.
2. Класи для реалізації розпаралелення програми.
3. Переваги та недоліки розпаралелення.

### **Практичне заняття 14**

#### **Тема 14. Потоки.**

##### **План практичного заняття**

1. Поняття потоку.
2. Класи для реалізації розпаралелення програми.
3. Переваги та недоліки розпаралелення.

## **6. САМОСТІЙНА РОБОТА**

<b>№ з/п</b>	<b>Назва теми</b>
1	Прості та складні програмні продукти. Їх особливість.
2	Об'єктно-орієнтоване програмування.
3	Парадигми програмування .
4	Поняття класу та екземпляр класу (об'єкт).
5	Специфікатори доступу (private, protected, public).
6	Конструктори та деструктори класу.
7	Ієрархія класів.
8	Віртуальні базові класи.
9	Контейнеризація.
10	Особливості віртуальних функцій.
11	Візуальні компоненти графічного інтерфейсу користувача.

12	Кросплатформенні додатки з JavaFX
13	Мережеве програмування засобами Java.
14	Потоки.

## 7. МЕТОДИ НАВЧАННЯ

Навчальний процес передбачає проведення лекцій і практичних занять в тому числі в комп'ютерній лабораторії, самостійну роботу студентів, виконання індивідуальних і розрахунково-графічних задач під керівництвом викладача та самостійно. Мова викладання – українська.

У процесі викладання дисципліни «Мережеве програмування» використовуються такі методи навчання як:

Словесні (пояснення, лекція, інструктаж).

Наочні (ілюстрація, мультимедійні файли).

Практичні (самостійна робота на занятті та позааудиторна діяльність, виконання лабораторних робіт).

Методи стимулювання обов'язку та відповідальності у студентів: переконання у значущості навчання, вимоги, вправи з виконання вимог, програмоване опитування;

Методи стимулювання інтересу до навчання: навчальні дискусії;

За ступенем самостійної роботи студентів: методи взаємодії викладача та студента (бесіда, дискусія); методи самостійної роботи студентів (самостійна робота, письмова робота, самостійна робота під керівництвом викладача);

За джерелом інформації і сприймання навчальної інформації: словесні (розповідь, бесіда, лекція, пояснення); наочні (ілюстрація, демонстрація).

У навчальному процесі застосовуються: лекції, в тому числі із використанням засобів комп'ютерної техніки; практичні заняття; індивідуальні заняття і виконання самостійної роботи, виконання розрахунково-графічних завдань під керівництвом викладача і самостійно; робота в Інтернет, робота на платформах Zoom, Moodle.

## 8. ФОРМИ КОНТРОЛЮ

Рубіжна атестація № 1	Директорська контрольна робота (№ 2)	Середній бал (№ 3)	разом
25%	50%	25%	100%

## 9. КРИТЕРІЇ ОЦІНЮВАННЯ

### КРИТЕРІЇ ОЦІНЮВАННЯ НАВЧАЛЬНИХ ДОСЯГНЕНЬ СТУДЕНТІВ З НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Оцінка за 100-бальною шкалою / ECTS	Оцінка за національною шкалою	Критерії оцінювання навчальних досягнень здобувачів освіти	Рівень компетентності
90-100  A	<b>Відмінно</b>	Оцінюється завдання, що містить відповіді, в яких навчальний матеріал відтворюється в повному обсязі. Здобувач освіти активно працює протягом усього курсу і показує при цьому високий рівень володіння теоретичними знаннями і практичними вміннями, відповідь його правильна, обґрунтована, повна, логічна, містить аналіз, систематизацію, узагальнення навчального матеріалу, здатний висловити власне ставлення до альтернативних міркувань з конкретної проблеми, проявляє вміння здійснювати зв'язок теоретичних занять з практичними і реалізувати міжпредметні зв'язки. Практичне завдання виконане правильно, як з використанням типового алгоритму, так і за самостійно розробленим алгоритмом.	Високий (творчий) рівень
85-89  B	<b>Дуже добре</b>	Оцінюється завдання, що містить відповіді в яких відтворюється значна частина навчального матеріалу. Здобувач освіти виявляє достатній рівень володіння теоретичним матеріалом і практичним вмінням з несуттєвими неточностями, певною мірою може аналізувати матеріал, порівнювати та робити висновки з окремих	Достатній рівень

		питань навчального матеріалу.	
75-84 С	<b>Добре</b>	Оцінюється завдання, що містить відповіді, в яких відтворюється незначна частина навчального матеріалу. Здобувач освіти виявляє достатній рівень володіння теоретичним матеріалом і практичним вмінням з певними неточностями та недоліками, достатньо проявляє вміння самостійно та аргументовано викладати матеріал.	
65-74 D	<b>Задовільно</b>	Оцінюється завдання, що містить відповіді в яких виявляється середній рівень володіння теоретичними знаннями, розуміння навчального матеріалу, а також практичних вмінь. Здобувач освіти в цілому оволодів суттю питань з даної теми, намагається аналізувати факти та події, робити певні конкретні висновки, недостатньо проявляє вміння самостійно та аргументовано викладати матеріал, виявляє недостатній рівень поінформованості у практичному застосуванні.	Середній рівень
60-64 E	<b>Достатньо</b>	Оцінюється завдання, що містить відповіді, в яких відтворюються мінімальні знання і розуміння навчального матеріалу, проте недостатньо вони глибокі та осмислені. Здобувач освіти на заняттях поводить себе пасивно, відповідає лише за викликом викладача, дає неповні відповіді на запитання, частково аналізує навчальний матеріал, проте при цьому робить неконкретні та неточні висновки, виявляє низький рівень поінформованості у практичному застосуванні.	

<p>35-5 FX</p>	<p><b>Незадовільно</b></p>	<p>Оцінюється завдання, що не виконане або містить відповіді на рівні елементарного відтворення окремих фактів, елементів, об'єктів, фрагментів навчального матеріалу. Здобувач освіти відтворює незначну частину навчального матеріалу, викладає його уривчастими реченнями, безсистемно, відсутні висновки, узагальнення. Під час відповіді здобувач освіти допускає суттєві помилки, які не здатен виправити після коригуючих запитань, рівень володіння теоретичними знаннями і практичним вмінням не задовольняє мінімальні критерії. Обов'язкове складання повторного екзамену, заліку.</p>	<p>Низький рівень</p>
<p>1-34 F</p>	<p><b>Незадовільно</b></p>	<p>Оцінюється завдання, що не виконане, у здобувача освіти відсутні елементарні як теоретичні знання, так і практичні вміння. Обов'язкове повторне вивчення навчальної дисципліни.</p>	<p>Незадовільний рівень</p>

## 10. ШКАЛА ОЦІНЮВАННЯ

Оцінка за 100-бальною шкалою	Оцінка за національною шкалою		Оцінка за шкалою ECTS	Рівень компетентності
	Екзамен	Залік		
90-100	5 (відмінно)	Зараховано	A	Високий (творчий) рівень
85-89	4 (дуже добре)		B	Достатній рівень
75-84	4 (добре)		C	
65-74	3(задовільно)		D	Середній рівень
60-64	3 (достатньо)		E	
35-59 (незадовільний рівень)	2 (незадовільно з можливістю повторного складання екзамену)	Незараховано (з можливістю повторного складання заліку)	FX	Низький рівень
1-34	2 (незадовільно з обов'язковим повторним вивченням дисципліни)	Незараховано (з обов'язковим повторним вивченням дисципліни)	X	Незадовільний рівень

## 11. МАТЕРІАЛЬНО-ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

1. Тексти лекцій.
2. Завдання для практичних завдань.
3. Завдання для самостійної роботи.
4. Тести для перевірки знань студентів.
5. Засоби діагностики знань студентів.
6. Критерії оцінювання знань студентів.
7. Посібники.
8. Презентаційні матеріали.
9. Інтернет-джерела.

## 12. РЕКОМЕНДОВАНА ЛІТЕРАТУРА

### **Базова:**

1. Kevin Sahin. The Java Web Scraping Handbook, ScrapingBee (March 06, 2020), 2020,115.
2. Thomas W. Christopher, George K. Thiruvathukal, Java Performance: In-Depth Advice for Tuning and Programming Java 8, 11, and Beyond 2nd Edition, Prentice Hall; eBook (Loyola eCommons; Creative Commons Licensed), 2020, 432.
3. Arun Gupta, Docker for Java Developers, O'Reilly Media, Inc.; eBook (Compliments of NGINX), 2020, 65.
4. Charles Humble, Ben Evans, Alex Blewitt, Siben Nayak. The Java Garbage Collection Mini-Book, FreeCodeCamp (2021), 2021.
5. Jason Goodwin, Kevin Webber. Cloud-Native Applications in Java: Build microservice-based cloud-native applications that dynamically scale. O'Reilly Media O'Reilly Media, Inc.; eBook (Compliments of VMware), 2018, 406.
6. Richard Jones, Antony Hosking, Eliot Moss. The Garbage Collection Handbook ("International Perspectives on Science, Culture and Society") 1st Edition, Chapman and Hall/CRC; 1st edition (September 30, 2020), 2020, 520.
7. Vaskaran Sarcar Java Design Patterns: A Hands-On Experience with Real-World Examples 2nd ed. Edition, Apress; 2nd ed. edition (December 7, 2018), 2018, 533.
8. Mykel Kochenderfer, Tim Wheeler, and Kyle Wray. Algorithms for Decision Making, The MIT Press (August 2, 2022); eBook (Creative Commons Edition). 2022.704.

### **Допоміжна:**

1. Ankur Moitra. Algorithmic Aspects of Machine Learning, Cambridge University Press (2018); eBook (Draft for Version 2), 2018, 158.
2. James Aspnes Notes on Data Structures and Programming Techniques, Yale University (Creative Commons Attribution-ShareAlike 4.0 (CC BY-SA 4.0)) (2022),2022.528c.
3. Matthias Felleisen, Robert Bruce Findler, Matthew Flatt, Shriram Krishnamurthi How to Design Programs, second edition: An Introduction to Programming and Computing (The MIT Press) second edition 2018, 792.
4. Florian Jatton The Constitution of Algorithms: Ground-Truthing, Programming, Formulating. The MIT Press, 2021, 154.
5. Xinyu Liu. Elementary Algorithms. [GitHub.com](https://github.com), 2021, 503.
6. Granville Barnett, Luca Del Tongo, John Bullinaria. Data Structures and Algorithms: Annotated Reference with Examples. University of Birmingham, 2019, 112.