

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЗАХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

ЗАТВЕРДЖУЮ  
В. о. декана факультету комп'ютерних  
інформаційних технологій  
Ігор ЯКИМЕНКО  
“\_\_\_\_\_” \_\_\_\_\_ 2023 р.

ЗАТВЕРДЖУЮ  
В. о. проректора з науково-  
педагогічної роботи  
Віктор ОСТРОВЕРХОВ  
“\_\_\_\_\_” \_\_\_\_\_ 2023 р.

ЗАТВЕРДЖУЮ  
Директор навчально-наукового  
інституту новітніх освітніх технологій  
Святослав ПИТЕЛЬ  
“\_\_\_\_\_” \_\_\_\_\_ 2023 р.

**РОБОЧА ПРОГРАМА**  
з дисципліни «Технології програмування для КСМ»

*ступінь вищої освіти – бакалавр*

галузь знань – 12 “Інформаційні технології”

спеціальність – 123 “Комп'ютерна інженерія”

освітньо-професійна програма – „Комп'ютерна інженерія”

Кафедра комп'ютерної інженерії

Форма навчання	Курс	Семестр	Лекції (год.)	Лабораторні (год.)	ІРС (год.)	Тренінг, КПЗ (год.)	Самост. робота студ. (год.)	Разом (год.)	Залік (сем.)	Екз. (сем.)
Денна	4	8	40	40	5	10	55	150	-	8
Заочна	4	8	8	4	0	0	138	150	-	8

31.08.2023  
*[Signature]*

Тернопіль – ЗУНУ  
2023

Робоча програма складена на основі освітньо – професійної програми підготовки бакалавра галузі знань 12 “Інформаційні технології” спеціальності 123 “Комп’ютерна інженерія”, затвердженої Вченою радою ЗУНУ (№ 10 від 24 червня 2020 р.)

Робочу програму склав к.т.н., доцент кафедри КІ

Юрій БАТЬКО

Робоча програма затверджена на засіданні кафедри комп’ютерної інженерії, протокол №1 від 28 серпня 2023р.

Завідувач кафедри



Леся ДУБЧАК

Розглянуто та схвалено групою забезпечення спеціальності «Комп’ютерна інженерія», протокол №1 від 28 серпня 2023 р.

Голова ГЗС



Олег БЕРЕЗЬКИЙ

Гарант ОП



Леся ДУБЧАК

СТРУКТУРА РОБОЧОЇ ПРОГРАМИ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ  
" ТЕХНОЛОГІЇ ПРОГРАМУВАННЯ ДЛЯ КСМ "

**1. Опис дисципліни " Технології програмування для КСМ "**

Дисципліна - «Технології програмування для КСМ»	Галузь знань, спеціальність, ступінь вищої освіти	Характеристика навчальної дисципліни
Кількість кредитів – 5	галузь знань – 12 „Інформаційні технології”	Статус дисципліни – нормативна Мова навчання – українська
Кількість залікових модулів: 4	Спеціальність – 123 „Комп’ютерна інженерія”	Рік підготовки: <i>Денна</i> - 4, <i>Заочна</i> - 4 Семестр: <i>Денна</i> – 8 <i>Заочна</i> – 8
Кількість змістових модулів – 3	Ступінь вищої освіти - бакалавр	Лекції: <i>Денна</i> - 40 год., <i>Заочна</i> – 8 год.  Лабораторні заняття: <i>Денна</i> - 40 год. <i>Заочна</i> – 4 год.
Загальна кількість годин – <i>Денна</i> – 150 год., <i>Заочна</i> – 150 год.		Самостійна робота: <i>Денна</i> - 55 год. <i>Заочна</i> – 138 год. Тренінги, КПІЗ – 10 год.  Індивідуальна робота: <i>Денна</i> -5 год.
Тижневих годин: <i>Денна</i> : 8 семестр – 18,75 год., з них аудиторних - 10 год.		Вид підсумкового контролю <i>Денна</i> : 8 семестр – екзамен <i>Заочна</i> : 8 семестр – екзамен

## **2. Мета й завдання дисципліни " Технології програмування для КСМ "**

### **2.1. Мета вивчення дисципліни**

Програма та тематичний план дисципліни „ Технології програмування для КСМ ” орієнтовані на глибоке та ґрунтовне засвоєння студентами систематичних знань та практичних навичок використання сучасних програмно-апаратних засобів для проектування, реалізації, верифікації, оптимізації та опису програмних систем з подальшим використанням набутих навичок під час розв’язання актуальних завдань.

Метою дисципліни є формування у студентів теоретичних знань та практичних навичок з розробки програмного забезпечення від етапу формування вимог до створення працездатного та якісного продукту.

### **2.2. Завдання вивчення дисципліни**

Завданням вивчення дисципліни є вивчення науково-практичного інструментарію проектування прикладних програмних систем та їх складових за допомогою сучасних засобів автоматизованого проектування.

В результаті вивчення дисципліни студенти мають вміти проводити декомпозицію поставлених задач, проектувати загальну структуру програмних систем, визначати та застосовувати методи оптимізації, створювати повний комплекс проектно-конструкторської документації.

### **2.3. Найменування та опис компетентностей, формування котрих забезпечує вивчення дисципліни “ Технології програмування для КСМ ”:**

К28. Здатність розуміти та використовувати сучасні парадигми та підходи до розроблення, впровадження та супроводу програмних засобів для комп’ютерних систем та мереж.

К29. Здатність застосовувати знання про технології проектування програмних додатків для комп’ютерних систем та мереж.

### **2.4 Передумови для вивчення дисципліни**

Зазначена дисципліна включена до циклу дисциплін професійної підготовки за переліком програми. У структурно-логічній схемі навчання зазначена дисципліна розміщена на IV-му курсі. Вивчення курсу „ Технології програмування для КСМ ” передбачає наявність систематичних та ґрунтовних знань із суміжних курсів («Технології проектування комп’ютерних систем», «Мережеве програмування»), а також цілеспрямованої роботи на лекційних та лабораторних заняттях, самостійної роботи студентів.

### **2.5. Результати навчання.**

В результаті вивчення курсу „ Технології програмування для КСМ ” студенти повинні:

ПРН24. Вміти застосовувати сучасні методи, парадигми та технології на всіх етапах життєвого циклу програмного забезпечення для комп’ютерних систем

та мереж.

ПРН25. Здатність створювати та впроваджувати програмне забезпечення комп'ютерних систем та мереж.

### **3. Програма навчальної дисципліни „Технології програмування для КСМ”**

#### ***Змістовий модуль 1. Проектування та розробка ПЗ***

##### **Теми 1.** Основні поняття програмної інженерії

1. Програмне забезпечення. 2. Проектування ПЗ. 3. Фази проектування ПЗ.  
4. Програмний продукт. 5. Критерії успішності проекту.

Література: 1, 2, 3.

##### **Тема 2.** Процес створення ПЗ.

1. Методології проектування ПЗ. 2. Характеристики методологій проектування. 3. Стратегії конструювання ПЗ. 4. Адаптивність процесу конструювання. 5. Вибір методології проектування.

Література: 2, 3.

##### **Тема 3.** Міжнародні стандарти життєвого циклу ПЗ

1. Класична модель проектування ПЗ. 2. Каскадна модель проектування. 3. Прототипування. 4. Інкрементна модель. 5. Спіральна модель. 6. Інкрементна спіральна модель.

Література: 4, 7.

##### **Тема 4.** Принципи та методології реалізації ПЗ

1. Швидка розробка програм (RAD). 2. Rational Unified Process. 3. Гнучкі методології (Agile). 4. Екстремальне програмування, його методи. 5. Методологія SCRUM. 6. Технологічні підходи до проектування ПЗ.

Література: 4, 7.

##### **Тема 5.** Патерни програмування та приклади реалізації

1. Поняття патернів. 2. Породжуючі патерни. 3. Структурні патерни. 4. Поведінкові патерни.

Література: 4, 7.

##### **Тема 6.** Інженерія вимог до ПЗ

1. Вимоги, їх властивості, види і обмеження. 2. Розробка вимог, їх виявлення і аналіз. 3. Специфікація вимог. 4. Документування і організація вимог. 5. Зміна вимог. 6. Планування і управління вимогами. 7. Аналіз, специфікація, верифікація та валідація вимог до ПЗ. 8. Шаблони специфікації вимог. 9. Стандарти вимог.

Література: 2, 5.

##### **Тема 7.** Проектування архітектури ПЗ.

Основні поняття проектування ПЗ. Структурування систем. Моделі управління програм. Модульна декомпозиція. Проблемно-залежні архітектури ПЗ. Шаблони проектування ПЗ.

Література: 5, 8.

**Тема 8.** Проектування інтерфейсу користувача.

Інтерфейс користувача. Принципи проектування інтерфейсів користувача. Представлення інформації. Засоби підтримки користувача. Оцінювання інтерфейсу.

Література: 4, 9.

**Змістовий модуль 2. Моделювання ПЗ.**

**Тема 9.** Класичні методи аналізу і проектування ПЗ.

Методологія моделювання SADT, IDEF, DFD, ELM, ООАД. Мови моделювання. Методи структурного проектування. Діаграми потоків даних. Методи, орієнтовані на структури даних. Метод проектування Джексона. Особливості класичних методів.

Література: 1, 3.

**Тема 10.** Об'єктно-орієнтований аналіз і проектування.

Основні поняття об'єктного підходу. Об'єкти і класи об'єктів. Ієрархія класів. Процес об'єктно-орієнтованого проектування. Визначення об'єктів. Моделі архітектури. Модифікація системної архітектури.

Література: 2, 5.

**Тема 11.** Архітектура розподілених програмних систем.

Розподілені системи. Багатопроекторна архітектура. Архітектура клієнт/сервер. Архітектура розподілених об'єктів. CORBA.

Література: 1, 3, 6, 7.

**Змістовий модуль 3. Менеджмент**

**Тема 12.** Управління програмними проектами.

Задачі управління проектами. Трикутник обмежень. Ресурси в програмних проектах. Співробітники, процеси і ролі. Планування ресурсів. Менеджмент проекту. Особливості програмних проектів, їх задачі. Виправлення програмних дефектів. Етапи і віхи програмних проектів.

Література: 1, 2, 3.

**Тема 13.** Управління ризиками в програмних проектах.

Управління ризиками. Ідентифікація ризиків. Аналіз ризиків. Планування управління ризиками. Оцінка ризиків. Методи управління ризиками. Стандарти управління ризиками.

Література: 1, 2, 6.

**Тема 14.** Системи управління дефектами

Дефекти. Їх характеристики. Ідентифікатор, стан і зміст дефекту. Життєвий

цикл дефекту. Класифікація систем управління дефектами. Їх основні характеристики.

Література: 2, 3.

#### **Змістовий модуль 4. Методи забезпечення та контролю якості ПЗ**

##### **Тема 15. Контроль версій в програмних проектах**

Групова робота з проектом. Спільний доступ. Збереження історії модифікацій. Системи контролю версій. Типи систем контролю версій. Робота з системами контролю версій. Система Git, CVS, Subversion, SmartSVN.

Література: 1, 2.

##### **Тема 16. Збір і випуск ПЗ.**

Задачі збору і випуску проектів. Причини збору проектів. Проблеми збору проектів. Процедури і засоби збору. Утиліти збору. Неперервна інтеграція. Дії для забезпечення неперервної інтеграції. Умови запуску збору. Промислові засоби неперервної інтеграції.

Література: 1, 3, 8.

##### **Тема 17. Якість програмного забезпечення**

Тенденції індустрії розробки ПЗ. Якість ПЗ: метрики і стандарти якості ПЗ. Забезпечення якості ПЗ. Методи забезпечення якості. Верифікація та валідація ПЗ. Тестування ПЗ.

Література: 3, 5.

##### **Тема 18. Інструментальні засоби інженерії програмного забезпечення.**

Інтегровані середовища розробки ПЗ. Системи управління проектами. Системи управління версіями документів, архітектурні особливості. Інструменти автоматизації зборки проектів. Інструменти автоматизації процесів тестування. Сервери постійної інтеграції.

Література: 1, 10.

### **4. Структура залікового кредиту дисципліни „Технології програмування для КСМ”.**

**(денна форма навчання)**

	Кількість годин					
	Лекції	Лабораторні заняття	Самостійна робота	Індивід. робота	Тренінг, КПЗ	Контрольні заходи
<b>Змістовий модуль 1</b>						
Теми 1. Основні поняття програмної інженерії	2	-	2	-	2	Опитування
Тема 2. Процес створення ПЗ.	2	6	4	1		Опитування
Тема 3. Міжнародні стандарти життєвого циклу ПЗ	4	-	4	1		Опитування
Тема 4. Принципи та методології реалізації ПЗ	2	-	4	-		Опитування

Тема 5. Патерни програмування та приклади реалізації	4	6	4	1		Опитування
Тема 6. Інженерія вимог до ПЗ	2	-	4	-		Опитування
Тема 7. Проектування архітектури ПЗ.	2	-	4	-		Опитування
Тема 8. Проектування інтерфейсу користувача.	2	6	4	-		Опитування
<b>Змістовий модуль 2</b>						
Тема 9. Класичні методи аналізу і проектування ПЗ.	2	-	4	1	4	Опитування
Тема 10. Об'єктно-орієнтований аналіз і проектування.	2	6	4	-		Опитування
Тема 11. Архітектура розподілених програмних систем.	2	-	2	-		Опитування
Тема 12. Управління програмними проектами.	2	-	2	1		Опитування
Тема 13. Управління ризиками в програмних проектах.	2	-	2	-		Опитування
Тема 14. Системи управління дефектами	2	-	2	-		Опитування
<b>Змістовий модуль 3</b>						
Тема 15. Контроль версій в програмних проектах	2	-	2	-	4	Опитування
Тема 16. Збір і випуск ПЗ.	2	8	2	-		Опитування
Тема 17. Якість програмного забезпечення	2	8	2	-		Опитування
Тема 18. Інструментальні засоби інженерії програмного забезпечення.	2	-	3	-		Опитування
Разом	40	40	55	5	10	

**(заочна форма навчання)**

	Кількість годин				
	Лекції	Лабораторні заняття	Індивідуальна робота	Тренінг, КПЗ	Самостійна робота
<b>Змістовий модуль 1</b>					
Теми 1. Основні поняття програмної інженерії	-	-	-	-	6
Тема 2. Процес створення ПЗ.	2	-	-	-	8
Тема 3. Міжнародні стандарти життєвого циклу ПЗ	-	-	-	-	6
Тема 4. Принципи та методології реалізації ПЗ	-	-	-	-	8
Тема 5. Патерни програмування та приклади реалізації	2	-	-	-	8
Тема 6. Інженерія вимог до ПЗ	-	-	-	-	8
Тема 7. Проектування архітектури ПЗ.	-	2	-	-	8
Тема 8. Проектування інтерфейсу користувача.	-	-	-	-	8
<b>Змістовий модуль 2</b>					
Тема 9. Класичні методи аналізу і проектування ПЗ.	-	-	-	-	6
Тема 10. Об'єктно-орієнтований аналіз і	-	-	-	-	8



проектування.					
Тема 11. Архітектура розподілених програмних систем.	2	-	-	-	8
Тема 12. Управління програмними проектами.	-	-	-	-	8
Тема 13. Управління ризиками в програмних проектах.	-	-	-	-	8
Тема 14. Системи управління дефектами	-	-	-	-	8
<b>Змістовий модуль 3</b>					
Тема 15. Контроль версій в програмних проектах	2	-	-	-	8
Тема 16. Збір і випуск ПЗ.	-	2	-	-	8
Тема 17. Якість програмного забезпечення	-	-	-	-	8
Тема 18. Інструментальні засоби інженерії програмного забезпечення.	-	-	-	-	8
Разом	8	4	0	0	138

## 5. Тематика лабораторних занять

### Лабораторна робота №1.

**Тема:** Технічне завдання для побудови програмної системи

**Мета:** Отримати теоретичні знання та практичні навички по розробці технічного завдання програмної системи

1. Поняття технічного завдання;
2. Стандарти та основні пункти технічного завдання
3. Переваги та недоліки технічного завдання

Література: 2, 4.

### Лабораторна робота №2.

**Тема:** Аналіз предметної області

**Мета:** Отримати теоретичні знання та практичні навички при аналізі технічного завдання, алгоритмів та програмно-апаратних аналогів при проектуванні програмних систем

**Питання для обговорення:**

1. Поняття предметної області;
2. Аналіз відомих алгоритмів та підходів
3. Оцінювання алгоритмів

Література: 2, 4, 5.

### Лабораторна робота №3.

**Тема:** Програмна система Visual Paradigm for UML

**Мета:** Отримати теоретичні знання та практичні навички при роботі з CASE – системою для моделювання програм Visual Paradigm for UML

**Питання для обговорення:**

1. Відношення залежності (dependency relationship);

2. Відношення асоціації (association relationship);
3. Відношення узагальнення (generalization relationship);
4. Відношення реалізації (realization relationship).

Література: 2, 5.

#### **Лабораторна робота №4.**

**Тема:** UML діаграм концептуального рівня моделювання архітектури програмної системи

**Мета:** Отримати теоретичні знання та практичні навички з методикою побудови UML-діаграм концептуального рівня проектування програмного забезпечення

##### **Питання для обговорення:**

1. Діаграми варіантів використання (або прецедентів – use case diagram);
2. Діаграми послідовностей (sequence diagram);
3. Діаграми пакетів (package diagram).

Література: 2, 5.

#### **Лабораторна робота №5.**

**Тема:** UML діаграми для проектування компонентних програмних рішень

**Мета:** Отримати теоретичні знання та практичні навички з методикою побудови UML-діаграм для проектування компонентних програмних рішень та моделювання динамічних аспектів

##### **Питання для обговорення:**

1. Діаграми кінцевого автомату (state machine diagram);
2. Діаграми діяльності (activity diagram);
3. Діаграми комунікації (communication diagram).

Література: 2, 4, 5.

#### **Лабораторна робота №6.**

**Тема:** Тестування програмної системи

**Мета:** Отримати теоретичні знання та практичні навички при проведенні комплексу завдань при дослідженні та тестуванні програмної системи

##### **Питання для обговорення:**

1. Алгоритми ручного тестування програмного забезпечення.
2. Алгоритми автоматизованого тестування програмного забезпечення.
3. Створення тест наборів.

Література: 2, 5, 7.

### **6. Комплексне практичне індивідуальне завдання**

Комплексні практичні індивідуальні завдання (КПІЗ) виконуються самостійно кожним студентом згідно з варіантом, отриманим у викладача. Варіанти КПІЗ з дисципліни „Технології програмування для КСМ”:

1. Комп’ютерна система розпізнавання статичних зображень
2. Комп’ютерна система аналізу відеопотоків

3. Комп'ютерна система аналізу змін напруги в електричній мережі
4. Комп'ютерна система документообігу в бібліотеці
5. Комп'ютерна система перевірки відвідування студентами занять на основі безконтактних карток
6. Комп'ютерна система інвентаризації товарів в комп'ютерному магазині на основі штрих-кодів
7. Комп'ютерна система обробки звукової інформації
8. Комп'ютерна система проти пожежної охорони для приміщення
9. Комп'ютерна система віртуального робочого місця для трейдера електронних бірж
10. Комп'ютерна система розпізнавання автомобільних знаків
11. Комп'ютерна система аналізу повітря на вміст шкідливих газів
12. Комп'ютерна система аналізу води на вміст шкідливих речовин
13. Комп'ютерна система обробки графічних зображень великого розміру
14. Комп'ютерна система контролю тиску в резервуарах великого розміру
15. Комп'ютерна система для моделювання 3-D об'єктів
16. Комп'ютерна система для верстки документів
17. Комп'ютерна система для створення анімаційних документів
18. Комп'ютерна система аналізу друкованих текстів на плагіат
19. Комп'ютерна система автоматичного детектування обличч
20. Комп'ютерна система для діагностики персонального комп'ютера.

## 7. Самостійна робота студентів

### (денна форма навчання)

№ п/п	Тематика
1	Історія розвитку програмної інженерії.
2	Розвиток модульного програмування.
3	Розвиток структурного програмування.
4	Основні положення об'єктного програмування.
5	Методи швидкої розробки програм.
6	Інструментальні засоби програмної інженерії.
7	Засоби тестування і наладки програм.
8	Комп'ютерна підтримка аналізу вимог до програмних виробів.
9	Інструментальне забезпечення життєвого циклу програм.
10	Мови розробки вимог до програмного виробу.
11	Мови підтримки структурного програмування.
12	Мови підтримки об'єктно-орієнтованого програмування.
13	Візуальні засоби в програмній інженерії.
14	Основні елементи графічного інтерфейсу користувача.
15	Основні етапи впровадження програмного забезпечення.

### (заочна форма навчання)

№ п/п	Тематика
1	Історія розвитку програмної інженерії.

2	Розвиток модульного програмування.
3	Розвиток структурного програмування.
4	Основні положення об'єктного програмування.
5	Методи швидкої розробки програм.
6	Інструментальні засоби програмної інженерії.
7	Засоби тестування і наладки програм.
8	Комп'ютерна підтримка аналізу вимог до програмних виробів.
9	Інструментальне забезпечення життєвого циклу програм.
10	Мови розробки вимог до програмного виробу.
11	Мови підтримки структурного програмування.
12	Мови підтримки об'єктно-орієнтованого програмування.
13	Візуальні засоби в програмній інженерії.
14	Основні елементи графічного інтерфейсу користувача.
15	Основи розробки розподілених програмних систем.
16	Основні елементи розробки програм реального часу.
17	Методи тестування і верифікації програмних систем.
18	Технічні засоби підтримки супроводу програмного забезпечення.
19	Основні етапи впровадження програмного забезпечення.
20	Перспективи розвитку методів програмної інженерії.

## **8 Організація і проведення тренінгу з дисципліни «Технології програмування для КСМ»**

<b>№п/п</b>	<b>Вид роботи</b>	<b>Порядок проведення тренінгу</b>
1	Огляд сучасних середовищ моделювання комп'ютерних програм	1. розгляд сучасних середовищ моделювання програм фірм Visual Paradigm, Microsoft; 2. дослідження мови графічного опису для об'єктного моделювання на основі UML діаграм
2	Розгляд процесу моделювання комп'ютерних програм	<ul style="list-style-type: none"> <li>• постановка задачі;</li> <li>• опис технічного завдання;</li> <li>• моделювання програмної розробки на основі UML діаграм</li> </ul>
3	Програмна реалізація проєктованого додатку	1. Реалізація розробленої програми на об'єктно-орієнтованій мові програмування; 2. реалізація графічного інтерфейсу користувача.
4	Тестування розробленого програмного додатку	<ul style="list-style-type: none"> <li>– вибір та обґрунтування тестової вибірки;</li> <li>– опис вхідних та вихідних даних;</li> <li>– перевірка правильності роботи реалізованого програмного додатку</li> </ul>

## **9. Засоби оцінювання та методи демонстрування результатів навчання**

У навчальному процесі застосовуються: лекції, в тому числі з використанням мультимедіапроектора та інших ТЗН; практичні заняття; індивідуальні заняття, самостійна робота студента, робота в Інтернет.

У процесі вивчення дисципліни „ Технології програмування для КСМ ” використовуються наступні засоби оцінювання та методи демонстрування результатів навчання:

- поточні опитування;
- залікове модульне тестування та опитування;
- презентації результатів виконання завдань та досліджень;
- оцінювання результатів КППЗ;
- завдання на лабораторному обладнанні;
- ректорська контрольна робота;
- екзамен.

### 10. Критерії, форми поточного та підсумкового контролю

В процесі вивчення дисципліни «Технології програмування для КСМ» використовуються наступні методи оцінювання навчальної роботи студента:

- поточне тестування та опитування;
- підсумкове тестування по кожному змістовому модулю;
- ректорська контрольна робота;
- підсумкова оцінка за комплексне практичне індивідуальне завдання (КППЗ), враховуючи поточне опитування;
- підсумковий письмовий екзамен.

Підсумковий бал (за 100-бальною шкалою) з дисципліни «Технології програмування для КСМ» визначається як середньозважена величина, в залежності від питомої ваги кожної складової залікового кредиту:

#### 8 семестр

Заліковий модуль 1	Заліковий модуль 2	Заліковий модуль 3 (підсумкова оцінка за КППЗ)	Заліковий модуль 4 (екзамен)	Разом
20 %	20 %	20 %	40 %	100%
1. Усне опитування під час заняття (8 тем по 5 бали = 40 балів) 2. Письмова робота = 60 балів	1. Усне опитування під час заняття (10 тем по 5 бали = 50 балів) 2. Письмова робота = 50 балів	1. Написання та захист КППЗ = 80 балів. 2. Виконання завдань під час тренінгу = 20 балів	1. Тестові завдання (25 тестів по 2 бали за тест) – макс. 50 балів 2. Завдання. 1 – макс. 25 балів 3. Завдання. 2 – макс. 25 балів	100

### Шкала оцінювання

За шкалою університету	За національною шкалою	За шкалою ECTS
90-100	Відмінно	A (відмінно)
85-89	Добре	B (дуже добре)
75-84		C (добре)
65-74	Задовільно	D (задовільно)
60-64		E (достатньо)
35-59	Незадовільно	FX (незадовільно, з можливістю повторного складання)
1-34		F (незадовільно, з обов'язковим повторним курсом)

## 11. Інструменти, обладнання та програмне забезпечення, використання яких передбачає навчальна дисципліна

№	Найменування	Номер теми
1.	Visual Studio Community Edition	1-18
2.	Visual Paradigm Community Edition	10-16

### РЕКОМЕНДОВАНІ ДЖЕРЕЛА ІНФОРМАЦІЇ

1. Alexander Stepanov and Paul McJones Elements of Programming Semigroup Press, 2019, 218
2. Kenneth Leroy Busbee, Dave Braunschweig. Programming Fundamentals - A Modular Structured Approach, 2nd Edition Houston Community College; eBook (Creative Commons Licensed)
3. Johan Sannemo Principles of Algorithmic Problem Solving KTH Royal Institute of Technology, 2018, 351.
4. Divakar Viswanath Scientific Programming and Computer Architecture. The MIT Press, 2017, 434.
5. Frank B. Brokken C++ Annotations: An Extensive Tutorial about the C++ Programming Language. University of Groningen, 2021.
6. Bjarne Stroustrup, Herb Sutter C++ Core Guidelines Standard C++ Foundation and GitHub. 2022.
7. 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.
8. 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.
9. 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.
10. Mykel Kochenderfer, Tim Wheeler, and Kyle Wray. Algorithms for Decision Making, The MIT Press (August 2, 2022); eBook (Creative Commons Edition). 2022. 704.
11. James Aspnes Notes on Data Structures and Programming Techniques, Yale University (Creative Commons Attribution-ShareAlike 4.0 (CC BY-SA 4.0)) (2022), 2022. 528 с.
12. Florian Jatton The Constitution of Algorithms: Ground-Truthing, Programming, Formulating. The MIT Press, 2021, 154.
13. Xinyu Liu. Elementary Algorithms. GitHub.com, 2021, 503.
14. Granville Barnett, Luca Del Tongo, John Bullinaria. Data Structures and Algorithms: Annotated Reference with Examples. University of Birmingham, 2019, 112.