



Основи програмування. Частина 2. Методології програмування

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Перший (бакалаврський)</i>
Галузь знань	<i>F Інформаційні технології</i>
Спеціальність	<i>F2 Інженерія програмного забезпечення</i>
Освітня програма	<i>Інженерія програмного забезпечення інформаційних систем</i>
Статус дисципліни	<i>Нормативна (цикл професійної підготовки)</i>
Форма навчання	<i>очна</i>
Рік підготовки, семестр	<i>1 курс, весняний семестр</i>
Обсяг дисципліни	<i>6 кредитів 180 годин (30 годин – лекції, 30 годин – лабораторні роботи. 120 годин – СРС)</i>
Семестровий контроль/ контрольні заходи	<i>Екзамен/ лабораторні роботи, модульна контрольна робота/</i>
Розклад занять	<i>http://rozklad.kpi.ua/Schedules/ScheduleGroupSelection.aspx</i>
Мова викладання	<i>Українська</i>
Інформація про керівника курсу / викладачів	<i>Лектор: к.т.н., Крамар Юлія Михайлівна, kramar.yuliia@edu.kpi.ua, моб. +38(050)3420-99-10 Лабораторні: старший викладач Вітківська Ірина Іванівна, iryna.vitkovska-fiot@lll.kpi.ua , моб. +38(067)598-31-89</i>
Розміщення курсу	<i>https://classroom.google.com/c/ODOxNTgzNjg2ODM0?cjc=2zzv43iq доступ за запрошенням викладача</i>

Програма навчальної дисципліни

1. Опис навчальної дисципліни, її мета, предмет вивчення та результати навчання

Навчальна дисципліна "Основи програмування" загалом і кредитний модуль "Основи програмування - 2. Методології програмування" зокрема, спрямовані на вивчення теоретичних та методологічних основ побудови програм на мовах програмування високого рівня з урахуванням сучасних концепцій і тенденцій розвитку, оволодіння інструментальними засобами створення таких програм, отримання практичних навичок розробки програмного забезпечення при вирішенні прикладних задач різного ступеня складності.

Відповідна теоретична та практична підготовка формує базові професійні навички з програмування і є основою для успішного опанування інших фахових дисциплін.

Метою дисципліни (кредитного модуля) є формування у студентів здатності розробляти програмне забезпечення для вирішення прикладних задач різного ступеня складності мовою програмування C# з використанням методології об'єктно-орієнтованого програмування.

Предмет навчальної дисципліни (кредитного модуля) – парадигма об'єктно-орієнтованого програмування, мова програмування високого рівня (C#), інструментальні засоби створення програмного забезпечення мовою C#.

Основні завдання кредитного модулю:

знання:

основних принципів об'єктно-орієнтованого програмування;
синтаксичних конструкцій і мовних засобів написання програм на мові C#;
базових принципів об'єктно-орієнтованого проектування.

вміння:

обирати за застосувати синтаксичні засоби мови C# при написанні коду;
використовувати об'єктно-орієнтовану парадигму, складаючи структуру коду;
обґрунтовувати обрані рішення об'єктно-орієнтованого проектування.

досвід:

проектування нескладних моделей предметної галузі, використовуючи об'єктно-орієнтований підхід, та програмування їх на мові C#.

Вивчення кредитного модуля «Основи програмування - 2. Методології програмування» сприяє формуванню у студентів наступних компетентностей.

Загальні компетентності:

- ЗК1. Здатність до абстрактного мислення, аналізу та синтезу.
- ЗК2. Здатність застосовувати знання у практичних ситуаціях.
- ЗК6. Здатність до пошуку, оброблення та узагальнення інформації з різних джерел.

Фахові компетентності, формуванню яких сприяє даний кредитний модуль:

- ФК1. Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення.
- ФК2. Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.
- ФК3. Здатність розробляти архітектури, модулі та компоненти програмних систем
- ФК7. Володіння знаннями про інформаційні моделі даних, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних.
- ФК8. Здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення.
- ФК10. Здатність накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя.
- ФК11. Здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення.
- ФК13. Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення.

ФК14. Здатність до алгоритмічного та логічного мислення.

Формування зазначених компетентностей забезпечує досягнення наступних програмних результатів навчання:

ПРН1. Аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки.

ПРН3. Знати основні процеси, фази та ітерації життєвого циклу програмного забезпечення.

ПРН6. Уміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення.

ПРН7. Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення.

ПРН13. Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань.

ПРН15. Мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення.

ПРН18. Знати та вміти застосовувати інформаційні технології обробки, зберігання та передачі даних.

2. Пререквізити та постреквізити дисципліни (місце в структурно-логічній схемі навчання за відповідною освітньою програмою)

Для успішного засвоєння дисципліни необхідно мати базові знання з кредитного модуля «Основи програмування - 1. Базові конструкції», кредитного модуля «Алгоритми і структури даних - 1. Основи алгоритмізації». Успішне оволодіння знаннями з дисципліни готує студентів до вивчення наступних дисциплін, таких як:

- «Компоненти програмної інженерії» (ПО 7),
- «Системне програмне забезпечення» (ПО 13),
- «Програмування веб-застосунків» (ПО 14),
- «Мультипарадигмене програмування» (ПО18),
- «Технології паралельних обчислень» (ПО 21).

3. Зміст навчальної дисципліни

Розділ 1. Парадигма об'єктно-орієнтованого програмування. Система типів C#. Клас і його складові

Тема 1.1 Еволюція програмування. Процедурне та об'єктно-орієнтоване програмування. Мови програмування.

Основні терміни та парадигми об'єктно-орієнтованого програмування. Складові процесу розробки програмного забезпечення. Принципи ООП. Класи. Об'єкти. Екземпляри та статичні члени класу.

Тема 1.2 Призначення методів в ООП. Способи передачі параметрів в C#. Повертання значень із методів.

Тема 1.3 Система типів в C#. Конструктори. Деструктори. Динамічний розподіл пам'яті в C#. Память стек та купа. Прибирання сміття в .Net.

Тема 1.4 Члени класу. Індикатори та властивості. Створення екземплярів об'єктів за допомогою властивостей. Правила використання властивостей

Тема 1.5 Перевантаження підпрограм (методів). Перевантаження унарних, бінарних, операцій, операторів приведення типів.

Розділ 2. Успадкування. Поліморфізм.

Тема 2.1 Успадкування, загальне положення. Методології декомпозиції. Одиночне успадкування в C#. Виклик конструктора базового класу. Порядок виконання конструкторів. Визначення членів, які успадковуються. Приведення типів в ієрархії успадкування.

Тема 2.2 Поліморфізм загальні положення. Раннє, пізнє зв'язування. Присвоєння та посилання на класи в ієрархії успадкування мовою C#. Приведення типів в ієрархії успадкування..

Тема 2.3 Віртуальні функції. Пізнє зв'язування. Реалізація поліморфізму мовою C#. Абстрактні класи та методи. Абстрактні методи в C#. Оператори `is` та `as`.

Тема 2.4 Інтерфейс. Порівняння абстрактних класів та інтерфейсів. Явна та неявна реалізація інтерфейсів

Тема 2.5 Відношення між класами та об'єктами. Порівняння та випадки застосування агрегації та композиції, агрегації та успадкування, успадкування та реалізації.

Розділ 3. Виключення.

Тема 3.1 Виключення. Оператори `try`, `catch`, `finally`. Виклик виключень. Оператор `throw`. Аргументи виключень. Конструктори класу `Exception`.

Вкладення блоків `try`. Повторна генерація виключень. Умови та правила генерації виключень. Стек виключення. Основні види виключень та їх призначення. Оператори `checked` `unchecked`. Власні виключення.

Розділ 4. Делегати. Обробка подій. Лямбда-вирази.

Тема 4.1 Об'явлення та використання делегатів. Обробка подій. Анонімні методи. Лямбда-вирази. Використання лямбда-виразів з делегатами.

Тема 4.2 Використання методів розширення та виразів запитів LINQ. Введення у запити LINQ. Мова запитів LINQ. LINQ to Objects. Операції LINQ.

Розділ 5. Структури даних. Лінійні та нелінійні структури даних.

Тема 5.1. Поняття структури даних. Лінійні та нелінійні структури даних. Прості спискові структури даних: стек, черга, дека. Складені спискові структури: двонаправлені списки, кільцеві списки, асоціативні списки.

Тема 5.2 Структури даних у бібліотеках .Net Collections та Collections.Generic. Шаблон об'єктно-орієнтованого програмування. Ітератор у колекціях.

4. Навчальні матеріали та ресурси

Базова література

1. Svetlin Nakov & co. Fundamentals of Computer Programming with C# — Sofia, 2013. —1122 с. [Електронний ресурс]. – Режим доступу: https://www.introprogramming.info/wp-content/uploads/2013/07/Books/CSharpEn/Fundamentals-of-Computer-Programming-with-CSharp-Nakov-eBook-v2013.pdf?utm_source=chatgpt.com
2. Learn C#. Free courses, tutorials, videos, and more about learning the programming language C# [Електронний ресурс]. – Режим доступу: https://dotnet.microsoft.com/en-us/learn/csharp?utm_source=chatgpt.com
3. Simon Kendal Object Oriented Programming using C# — bookboon.com —254 с. [Електронний ресурс]. – Режим доступу: https://dvikan.no/ntnu-studentserver/kompendier/object-oriented-programming-using-c-sharp.pdf?utm_source=chatgpt.com

Додаткова література

1. N. Wirth. Algorithms and Data Structures. — Oberon, 2004. — 360 p.
2. Joe Albahari C# 12 in a Nutshell: The Definitive Reference — O'Reilly Media, 2023. —1084 с.
3. Jeffrey Richter CLR via C# (Developer Reference) 4th Edition — Microsoft Press, 2012. — 896 с.
4. Головна сторінка середовища MS Visual Studio [Електронний ресурс]. – Режим доступу: <http://visualstudio.com>
5. Головна сторінка MSDN [Електронний ресурс]. – Режим доступу: <http://msdn.microsoft.com>

5. Навчальний контент

5.1. Лекційні заняття

Матеріали для вивчення дисципліни розміщені в електронному вигляді на сайті <https://learn.epam.com> та в гугл-класі, посилання надаються студентам на початку семестру. Контент доступний студенту із будь-якого місця в мережі Інтернет.

Лекції по дисципліні проводяться із використанням сучасних мультимедійних презентаційних технологій.

Таблиця 1

Лекційні заняття

Лекції	Назва теми лекції та перелік основних питань
	Розділ 1. Парадигма об'єктно-орієнтованого програмування. Система типів C#. Клас і його складові

1	Тема 1.1 Еволюція програмування. Процедурне та об'єктно-орієнтоване програмування. Мови програмування. Основні терміни та парадигми об'єктно-орієнтованого програмування. Складові процесу розробки програмного забезпечення. Принципи ООП. Класи. Об'єкти. Екземпляри та статичні члени класу
2	Тема 1.2 Призначення методів в ООП. Способи передачі параметрів в C#. Повертання значень із методів.
3	Тема 1.3 Система типів в C#. Конструктори. Деструктори. Динамічний розподіл пам'яті в C#. Пам'ять стек та купа. Прибирання сміття в .Net.
4	Тема 1.4 Члени класу. Індексатори. Властивості. Автоматичні властивості. Створення екземплярів об'єктів за допомогою властивостей.
5	Тема 1.5 Перевантаження підпрограми (методів). Перевантаження унарних та бінарних операцій.
	Розділ 2. Успадкування. Поліморфізм.
6	Тема 2.1 Успадкування загальне положення. Методології декомпозиції. Одиночне успадкування в C#. Виклик конструктора базового класу. Порядок виконання конструкторів. Визначення членів, які успадковуються. Приведення типів в ієрархії успадкування. Правила успадкування.
7	Тема 2.2 Поліморфізм загальні положення. Раннє, пізнє зв'язування. Присвоєння та посилання на класи в ієрархії успадкування мовою C#. Приведення типів в ієрархії успадкування.
8	Тема 2.3 Віртуальні функції. Пізнє зв'язування. Реалізація поліморфізму в C#. Абстрактні класи та методи. Абстрактні методи в C#. Оператори is та as.
9	Тема 2.4 Інтерфейс. Порівняння абстрактних класів та інтерфейсів. Явна та неявна реалізація інтерфейсів.
10	Тема 2.5 Відношення між класами та об'єктами. Порівняння та випадки застосування агрегації та композиції, агрегації та успадкування, успадкування та реалізації.
	Розділ 3. Виключення.
11	Тема 3.1 Виключення. Оператори try, catch, finally. Виклик виключень. Оператор throw. Аргументи виключень. Конструктори класу Exception. Вкладення блоків try. Повторна генерація виключень. Умови та правила генерації виключень. Стек виключення. Основні види виключень та їх призначення. Оператори checked unchecked. Власні виключення.
	Розділ 4. Делегати. Обробка подій. Лямбда-вирази.
12	Тема 4.1 Об'явлення та використання делегатів. Обробка подій. Анонімні методи. Лямбда-вирази. Використання лямбда-виразів з делегатами.

13	Тема 4.2 Використання методів розширення та виразів запитів LINQ. Введення у запити LINQ. Мова запитів LINQ. LINQ to Objects. Операції LINQ.
Розділ 5. Структури даних. Лінійні та нелінійні структури даних.	
14	Тема 5.1. Поняття структури даних. Лінійні структури даних. Прості спискові структури даних: стек, черга, дека. Складені спискові структури: двонаправлені списки, кільцеві списки, асоціативні списки.
15	Тема 5.2. Структури даних у бібліотеках .Net Collections та Collections.Generic. Шаблон об'єктно-орієнтованого програмування. Ітератор у колекціях.

5.2. Лабораторні заняття

Лабораторні заняття виконуються з використанням мови програмування C# та версій середовищ розробки програм (IDE), які надаються розробниками для навчальних цілей безкоштовно. Вказівки до виконання лабораторних робіт включені в онлайн курс на платформі <https://learn.epam.com>. Практичні завдання в онлайн-курсі складають основу лабораторних робіт. Особливістю форми представлення практичних завдань онлайн-курсу є залучення спеціальної системи Autocode, яка дозволяє автоматизувати перевірку коду. Якщо студент виконав завдання, та Autocode оцінив позитивно код лабораторної роботи та стилістику його написання, студент захищає роботу на занятті з викладачем, відповідаючи на теоретичні запитання. Також викладач визначає академічну добродію студента та відсутність плагіату. Для виконання кожної лабораторної роботи студенту надається не більше 2 тижнів.

За умови навчання у форматі online заняття, у тому числі контрольні заходи, проводяться з використанням сервісу Google, MS Teams та цифрової платформи Learn.

Вихідний код лабораторних робіт має бути розміщеним кожним студентом особисто у глобальному репозиторії <https://autocode.git.epam.com>.

Таблиця 2

Лабораторні заняття

№ з/п	Назва лабораторної роботи (комп'ютерного практикуму)	Ресурс	Кількість ауд. годин
1	Функції та процедури	Autocode	3
2	Опис та використання класів	Autocode	3
3	Успадкування	Autocode	4
4	Поліморфізм	Autocode	4
5	Інтерфейси. Реалізація інтерфейсів	Autocode	4
6	Обробка виключень	Autocode	4
7	Побудова та використання структур даних	Метод.вказ.	4
8	Відношення між класами та об'єктами	Метод.вказ.	4

5.3. Самостійна робота студента

До самостійної роботи студента відноситься виконання індивідуальних завдань з тематики, яка виноситься на лабораторні роботи, а також опрацювання теоретичного матеріалу за наданими текстами лекцій та додатковою літературою, у тому числі за темами, які винесені на самостійне вивчення (згідно таблиці 1). Усі навчальні матеріали (тексти лекцій, методичні

вказівки до виконання лабораторних робіт) розміщені в електронному вигляді на сайті (<https://learn.epam.com/study/>), а також в гугл-класі. Навчальний контент доступний із будь-якого місця в мережі Інтернет.

На самостійну роботу студент має витратити кількість годин, що співмірна із кількістю годин, проведених ним на аудиторних заняттях.

Таблиця 3

Самостійна робота

№ з/п	Назва тем самостійної роботи	Год.
1	Тема 1.2 Опціональні та іменовані параметри, параметри-масиви та довільна кількість параметрів	8
2	Тема 1.3. Оптимізація роботи прибиральника сміття	8
3	Тема 1.4 Рекомендації використання властивостей та індексаторів	8
4	Тема 1.5 Перевантаження операторів приведення типів	8
5	Тема 2.1 Відкрите, захищене та закрите наслідування	8
6	Тема 2.2 Побічні ефекти множинного наслідування	8
7	Тема 2.3 Використання класу Object у бібліотечних класах	8
8	Тема 2.4 Перевизначення інтерфейсів	8
9	Тема 2.5 Використання відношень у шаблонах проектування – базові відомості та приклади	8
10	Тема 3.1 Аналіз стеку виклику виключення. Вкладене виключення	8
11	Тема 4.1 Використання подій в застосуваннях с графічним інтерфейсом.	8
12	Тема 4.2 Динамічний LINQ	8
13	Тема 5.1. Реалізація колізій хеш-таблиць	8
14	Тема 5.2 Додаткові типи колекцій: множина, відсортований словник	8
15	Тема 5.2. Структури даних у бібліотеках .Net Collections та Collections.Generic.	8
	Разом	120

6. Політика навчальної дисципліни (освітнього компонента) Як викладач, так і студент зобов'язані дотримуватись [Кодексу честі Національного технічного університету України «Київський політехнічний інститут»](#).

Основні положення політики:

- відвідування лекційних та лабораторних занять є обов'язковою (за можливості) складовою вивчення матеріалу;
- студент повинен вивчати дисципліну самостійно та вчасно;
- впродовж занять студенти можуть задавати питання стосовно матеріалу, що викладається; студенти мають можливість підняти будь-яке питання, яке стосується процедури проведення та оцінювання контрольних заходів;

- студенти мають право оскаржити результати контрольних заходів, аргументовано пояснивши з яким критерієм не погоджуються відповідно до критеріїв оцінки та/або зауважень;
- у випадку виявлення факту академічної недоброчесної роботи лабораторна робота не зараховується.

Відвідування є обов'язковим (за можливості) (за винятком випадків, коли існує поважна причина, наприклад, хвороба чи дозвіл працівників деканату). Якщо студент не може бути присутнім на заняттях, він все одно несе відповідальність за опрацювання теоретичного матеріалу та виконання лабораторних робіт згідно графіку.

Політика перескладань. Пропущені (за наявності поважних причин) роботи можна здати після дедлайну без втрати балів, але не пізніше останнього лабораторного заняття в семестрі і не більше одної роботи за раз. Якщо студент не набрав достатню кількість балів, то він може бути допущений до основної сесії (матиме багато додаткових питань на заліку), але не допущений до перескладань на додатковій сесії. Такі обмеження стимулюють студента організувати систематичне виконання завдань та не допускати значного накопичення незданих лабораторних робіт на кінець семестру. Перенесення вивчення дисципліни на наступний семестр як додаткової послуги неможливе.

Визнання результатів навчання, набутих у неформальній/інформальній освіті. Порядок визнання таких результатів регламентується Положенням про визнання результатів навчання, набутих у неформальній / інформальній освіті (<https://osvita.kpi.ua/index.php/node/179>). Можуть бути зараховані окремі змістовні модулі або теми дисципліни. В такому разі здобувач звільняється від виконання відповідних завдань, отримуючи за них максимальний бал відповідно до рейтингової системи оцінювання.

Політика академічної поведінки та доброчесності: конфліктні ситуації мають відкрито обговорюватись в академічних групах з викладачем, необхідно бути взаємно толерантним, поважати думку іншого. Плагіат та інші форми нечесної роботи неприпустимі. Всі завдання студент має виконати самостійно із використанням рекомендованої літератури й отриманих знань та навичок. Цитування в письмових роботах допускається тільки із відповідним посиланням на авторський текст. Недопустимі підказки і списування у ході захисту робіт, на тестах, на заліку.

Обов'язковою умовою виконання завдань з освітньої компоненти є дотримання політики та принципів академічної доброчесності (<https://kpi.ua/academic-integrity>), які, у тому числі, викладено у Кодексі честі Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» (<https://kpi.ua/code>), Положенні про систему запобігання академічному плагіату в КПІ ім. Ігоря Сікорського (<https://osvita.kpi.ua/node/47>). У разі виявлення дублювання робіт, плагіату роботи здобувачі отримують нульовий рейтинг.

Норми академічної етики: дисциплінованість; дотримання субординації; чесність; відповідальність; робота в аудиторії з відключеними мобільними телефонами. Повага один до одного дає можливість ефективніше досягати поставлених результатів. Якщо студент використовує свій ноутбук чи телефон для аудіо- чи відеозапису, необхідно заздалегідь отримати дозвіл викладача.

Дотримання академічної доброчесності студентів й викладачів регламентується кодексом честі Національного технічного університету України «Київський політехнічний інститут», положенням про організацію освітнього процесу в КПІ ім. Ігоря Сікорського. За порушення принципів академічної доброчесності, зокрема плагіат лабораторних робіт, студент втрачає всі бали за дану роботу. Студент має вивчати дисципліну протягом семестру, дотримуючись календарного плану. Усі завдання студент має виконувати самостійно і вчасно.

Політика використання штучного інтелекту. Використання штучного інтелекту (далі, ШІ) регламентується «Політикою використання штучного інтелекту для академічної діяльності в КПІ ім. Ігоря Сікорського» (<https://osvita.kpi.ua/node/1225>). Усі навчальні завдання з дисципліни мають бути результатом власної оригінальної роботи здобувача.

Для перевірки як теоретичних знань, так і практичних навичок в семестрі передбачена модульна контрольна робота (МКР), яка може складатися з двох частин, які проводяться під час семестрових атестацій.

Завдання та запитання модульної контрольної роботи – це тренування студента до запитань та завдань екзаменаційного білету. Дату та час проведення робіт оголошує лектор на початку семестру, відмічає її у журналі та нагадує про проведення роботи на лекційних заняттях. Переписати модульну контрольну роботу неможливо.

Оцінювання студентів здійснюється згідно рейтингової оцінки рівня підготовки студентів з дисципліни. Поточний стан успішності студенти можуть бачити в системі «Електронний кампус». Рейтингова система оцінювання з кредитного модуля описана у наступному розділі робочої програми.

7. Види контролю та рейтингова система оцінювання результатів навчання (PCO)

7.1. Поточний контроль

Поточний контроль успішності засвоєння знань виконується шляхом виконання ними: лабораторних робіт, МКР, завдань, що видані для самостійної проробки.

Семестровий рейтинг студента з дисципліни складається з балів, що він отримує за:

- лабораторні роботи;
- модульну контрольну роботу (МКР може поділяється на дві контрольні роботи тривалістю по одній академічній годині);
- іспит.

7.2. Система рейтингових (вагових) балів та критерії оцінювання

Лабораторні роботи

Вагові бали кожної лабораторної роботи наведені у Таблиці 4. Сумарний ваговий бал за даний контрольний захід (*r_z*) складає **42 бали**.

Критерії оцінювання лабораторних робіт включають якість її виконання та захисту (Таблиця 4)

Таблиця 4

Вагові бали та критерії оцінювання лабораторних робіт.

№	Назва роботи	Бали		
		Виконання	Захист	Сума
1	Функції та процедури	3		3
2	Опис та використання класів	3		3
3	Успадкування	3		3
4	Поліморфізм	3		3
5	Інтерфейси. Реалізація інтерфейсів	3		3
6	Обробка виключень	3		3
	Проміжний тестовий контроль по лабораторних роботах 1- 6	4		4
7	Побудова та використання структур даних	3	7	10
8	Відношення між класами та об'єктами	3	7	10
	Разом за лабораторні роботи	28	14	42

Критерії оцінювання лабораторних робіт 1-6 (оцінюються системою автоматичної перевірки завдань Autocode):

- виконане завдання з автоматичною перевіркою на Autocode – 3 бали;
- після виконання лабораторних робіт 1-6 виконується тестове завдання для сумарної перевірки теоретичних знань з тем робіт 1-6 – оцінка виконання тестового завдання вираховується наступним чином: 100% правильних відповідей тесту оцінюються в 4 бали. Процент неправильних відповідей віднімається відповідно від максимальної оцінки за тест. Проміжний тестовий контроль також відіграє роль тренування написання модульної контрольної роботи. Результат проміжного тестового контролю зраховується тільки після виконання 1-6 робіт;
- роботи 7-8 виконуються за індивідуальним варіантом, наданим викладачем, код демонструється викладачу і захищається на занятті (лабораторна робота зраховується студенту тільки при умові її захисту).

Критерії оцінювання лабораторних робіт 7- 8:

- **“відмінно”**, повна відповідь (не менше 95% потрібної інформації); - 10 балів;
- **“добре”**, достатньо повна відповідь (не менше 75% потрібної інформації, або незначні неточності); - 7-8 балів;
- **“задовільно”**, неповна відповідь (не менше 60% потрібної інформації та деякі помилки); - 6 балів;
- **“незадовільно”**, незадовільна відповідь. - < 6 балів

Модульна контрольна робота

Ваговий бал МКР –8 балів.

Критерії оцінювання кожної частини МКР:

- **“відмінно”**, повна відповідь (не менше 95% потрібної інформації); - 8 балів
- **“добре”**, достатньо повна відповідь (не менше 75% потрібної інформації, або незначні неточності); - 6 – 7 балів
- **“задовільно”**, неповна відповідь (не менше 60% потрібної інформації та деякі помилки); - 4 - 5 балів
- **“незадовільно”**, незадовільна відповідь. - < 4 балів

Умови позитивної проміжної атестації

Для отримання “зараховано” з першої проміжної атестації (7 тиждень) студент повинен мати не менше ніж 12 балів (за умови, якщо на початок 7 тижня згідно з календарним планом контрольних заходів “ідеальний” студент має отримати 15 балів).

Для отримання “зараховано” з другої проміжної атестації (13 тиждень) студент повинен мати не менше ніж 20 балів (за умови, якщо на початок 13 тижня згідно з календарним планом контрольних заходів “ідеальний” студент має отримати 32 балів).

Необхідною умовою допуску до екзамену є:

- написання тестового завдання проміжного контролю;
- виконання МКР не нижче ніж на оцінку “задовільно”;
- сума проміжних оцінок має складати не менше 37 балів.

Максимальна сума вагових балів контрольних заходів протягом семестру складає:

$$R_c = 42 + 8 = 50 \text{ балів}$$

Критерії екзаменаційного оцінювання:

Екзаменаційне завдання складається з двох частин: практичної та теоретичної. В практичній частині екзаменаційної роботи необхідно написати програмний код, використовуючи методологію об'єктно орієнтованого програмування, засновану на представленні програми у вигляді сукупності об'єктів, кожен з яких є екземпляром певного класу, які можуть утворювати ієрархію спадкування.

Критерії оцінювання якості практичної частини роботи:

- **“відмінно”**, повна відповідь (не менше 95% потрібної інформації); - 33-35 бали
- **“добре”**, достатньо повна відповідь (не менше 75% потрібної інформації, або незначні неточності); - 26-32 балів
- **“задовільно”**, неповна відповідь (не менше 60% потрібної інформації та деякі помилки); - 21-31 балів
- **“незадовільно”**, незадовільна відповідь. - < 20 балів

Ваговий бал теоретичної частини екзаменаційної роботи – 35 балів.

Критерії оцінювання якості теоретичної частини роботи:

- **“відмінно”**, повна відповідь (не менше 95% потрібної інформації); - 14.5 -15 бали
- **“добре”**, достатньо повна відповідь (не менше 75% потрібної інформації, або незначні неточності); - 11-13 балів
- **“задовільно”**, неповна відповідь (не менше 60% потрібної інформації та деякі помилки); - 9-10 балів
- **“незадовільно”**, незадовільна відповідь. - < 9 балів

Ваговий бал теоретичної частини екзаменаційної роботи – 15 балів.

Максимальна екзаменаційна складова шкали R_e дорівнює 50 балів, а саме:

- **“відмінно”** - 48-50 бали
- **“дуже добре”** - 43-47 балів
- **“добре”** - 38-42 балів
- **“задовільно”** - 33-37 балів
- **“достатньо”** - 30-32 балів
- **“незадовільно”** - < 30 балів

Таким чином, рейтингова шкала з дисципліни складає $R_c + R_e = 50 + 50 = 100$ балів.

Для отримання студентом відповідних оцінок - його рейтингова оцінка R_D переводиться в оцінку згідно з таблицею 5.

Таблиця 5

**Таблиця відповідності рейтингових балів
оцінкам за університетською шкалою**

Бали $R_D=R_C+R_E$	Оцінка
95... 100	відмінно
85 ... 94	дуже добре
75 ... 84	добре
65 ... 74	задовільно
60 ... 64	достатньо
$R_D < 60$	незадовільно
$r^C < 50$ або не виконані інші умови допуску до екзамену	Не допущений

Робочу програму навчальної дисципліни (силабус):

Складено доцентом, к.т.н, доцентом, Крамар Юлією Михайлівною, старшим викладачем Вітківською Іриною Іванівною.

Ухвалено кафедрою ІІІ (протокол №2/1 від 10.10.2025р.)

Погоджено Методичною комісією факультету (протокол №3 від 17.10.2025р.)