

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ МОРСЬКИЙ УНІВЕРСИТЕТ
КАФЕДРА КІБЕРБЕЗПЕКИ ТА ЗАХИСТУ ІНФОРМАЦІЇ**

**МЕТОДИЧНІ ВКАЗІВКИ
до виконання розрахунково-графічної роботи
з освітнього компонента (навчальної дисципліни)
Об'єктно-орієнтоване програмування
для здобувачів першого (бакалаврського) рівня вищої освіти
за спеціальністю
125 Кібербезпека та захист інформації
(частина 2)**

Одеса – 2025

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ МОРСЬКИЙ УНІВЕРСИТЕТ
КАФЕДРА КІБЕРБЕЗПЕКИ ТА ЗАХИСТУ ІНФОРМАЦІЇ**

**МЕТОДИЧНІ ВКАЗІВКИ
до виконання розрахунково-графічної роботи
з освітнього компонента (навчальної дисципліни)
Об'єктно-орієнтоване програмування
для здобувачів першого (бакалаврського) рівня вищої освіти
за спеціальністю
125 Кібербезпека та захист інформації
(частина 2)**

**Затверджено
на засіданні кафедри кібербезпеки та
захисту інформації
Протокол № 1 від 01.09.2025р.**

Одеса – 2025

Методичні вказівки до виконання розрахунково-графічної роботи з освітнього компонента (навчальної дисципліни) Об'єктно-орієнтоване програмування для здобувачів першого (бакалаврського) рівня вищої освіти за спеціальністю 125 Кібербезпека та захист інформації (частина 2) / Укл.: К.О. Трифонова. – Одеса, 2025. – 15 с.

Укладач: Трифонова К.О., ст. викл.

ЗМІСТ

Вступ.....	5
Завдання №9. Інтерфейси, структури і перерахування	6
Завдання №10. Система обробки виняткових ситуацій	7
Завдання №11. Стандартні та користувацькі винятки.....	7
Завдання №12. Делегати та анонімні функції	8
Завдання №13. Обробка подій	9
Завдання №14. Узагальнені класи	10
Завдання №15. Узагальнені типи.....	11
Завдання №16. Мова інтегрованих запитів.....	12
Завдання №17. Небезпечний код та покажчики.....	13

ВСТУП

Навчальна дисципліна Об'єктно-орієнтоване програмування є необхідною дисципліною для підвищення рівня теоретичних і прикладних знань, що формують фахівця в галузі інформаційних технологій, які сприяють утворенню у здобувачів поглиблених вмінь та навичок розробки програмного забезпечення із застосуванням сучасних парадигм програмування, тобто підготовці спеціалістів для створення, впровадження та підтримки професійно-орієнтованих комп'ютерних технологій у професійній діяльності.

Дисципліна Об'єктно-орієнтоване програмування відповідає освітньо-професійній програмі, навчальному плану підготовки фахівців першого (бакалаврського) рівня вищої освіти спеціальності 125 Кібербезпека та захист інформації і є складовою циклу дисциплін професійної підготовки обов'язкової частини навчального плану.

Дисципліну викладають впродовж третього семестру першого (бакалаврського) рівня. У процесі навчання передбачено лекції, практичні заняття.

Згідно навчального плану передбачено підсумковий контроль у вигляді екзамену.

Робочу програму навчальної дисципліни укладено згідно з вимогами кредитно-модульної системи організації навчального процесу. Програма визначає обсяги компетентностей, які повинен опанувати здобувач відповідно до освітньо-професійної програми, алгоритму вивчення навчального матеріалу дисципліни Об'єктно-орієнтоване програмування, необхідне методичне забезпечення, складові та технологію оцінювання навчальних досягнень.

Предмет навчальної дисципліни Об'єктно-орієнтоване програмування – сучасні технології розробки програмного забезпечення.

Мета вивчення навчальної дисципліни Об'єктно-орієнтоване програмування – забезпечити формування поглибленої системи теоретичних і практичних знань у галузі розробки програмного забезпечення для реалізації здатності використання інформаційно-комунікаційних технологій, сучасних методів і моделей інформаційної безпеки та/або кібербезпеки.

Завдання вивчення дисципліни:

- формувати теоретичні основи та практичні навички застосування методів вилучення, аналізу, обробки та синтезу інформації для формування технічного завдання при розв'язанні задач в галузі інформаційних технологій;

- формувати теоретичні основи та практичні навички сучасних парадигм програмування та способів їх вибору з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі інформаційних технологій;

- поглибити практичні навички кодування програмного забезпечення для розв'язання задач в галузі інформаційних технологій;

- поглибити практичні навички налагодження програмного забезпечення для розв'язання задач в галузі інформаційних технологій.

Стратегічні цілі дисципліни – націлити майбутніх фахівців на застосування отриманих знань у подальшій професійній підготовці та їх наступній практичній діяльності.

ЗАВДАННЯ №9. ІНТЕРФЕЙСИ, СТРУКТУРИ І ПЕРЕРАХУВАННЯ

1. Дайте визначення поняттю інтерфейс для програми на C#.
2. Назвіть основну відмінність інтерфейсу від абстрактного класу для програми на C#.
3. Назвіть кількість класів, в яких можна реалізувати один інтерфейс для програми на C#.
4. Поясніть неможливість наявності членів даних в інтерфейсі для програми на C#.
5. Назвіть модифікатор доступу, який за замовчуванням призначається кожному члену інтерфейсу для програми на C#.
6. Наведіть загальну форму оголошення інтерфейсу для програми на C#.
7. Перерахуйте, які методи є доступними змінній посилального інтерфейсного типу для програми на C#.
8. Наведіть загальну форму оголошення інтерфейсу з властивістю для програми на C#.
9. Наведіть загальну форму оголошення інтерфейсу з індексатором для програми на C#.
10. Поясніть особливості, що виникають при реалізації у класі похідного інтерфейсу для програми на C#.
11. Поясніть необхідність використання приховування імен при спадкуванні інтерфейсів для програми на C#.
12. Поясніть причини використання явної реалізації члена інтерфейсу для програми на C#.
13. Наведіть рекомендацію, у яких випадках слід використовувати інтерфейс, а у яких – абстрактний клас для програми на C#.
14. Перерахуйте декілька стандартних інтерфейсів, визначених для середовища .NET Framework, якими можна користуватися для програми на C#.
15. Наведіть загальну форму оголошення структури для програми на C#.
16. Поясніть головну відмінність структур від класів для програми на C#.
17. Наведіть загальну форму оголошення перерахування для програми на C#.
18. Поясніть, до якого типу даних відноситься кожна символічно позначена константа у перерахуванні для програми на C#.
19. Поясніть необхідність вказівки базового типу даних для перерахування для програми на C#.
20. Назвіть основну перевагу використання перерахування для програми на C#.

ЗАВДАННЯ №10. СИСТЕМА ОБРОБКИ ВИНЯТКОВИХ СИТУАЦІЙ

1. Дайте визначення поняттю винятку для програми на C#.
2. Дайте визначення системі обробки виняткових ситуацій для програми на C#.
3. Дайте визначення поняттю обробник винятків для програми на C#.
4. Поясніть призначення основних операторів при побудові системи обробки виняткових ситуацій для програми на C#.
5. Наведіть загальну форму використання блоків try/catch при побудові системи обробки виняткових ситуацій для програми на C#.
6. Поясніть порядок виконання блоків, які утворюють систему обробки виняткових ситуацій, у разі виникнення винятку виходу за межі масиву в Main() для програми на C#.
7. Поясніть порядок виконання блоків, які утворюють систему обробки виняткових ситуацій, у разі виникнення винятку виходу за межі масиву у методі, що викликається в Main() для програми на C#.
8. Назвіть перевагу використання системи обробки виняткових ситуацій для програми на C#.
9. Назвіть складову .Net Framework, яка виконує генерування вбудованих винятків для програми на C#.
10. Поясніть порядок виконання блоків, які утворюють систему обробки виняткових ситуацій у разі виникнення кількох однакових винятків для програми на C#.
11. Поясніть необхідність використання кількох операторів catch при побудові системи обробки виняткових ситуацій для програми на C#.
12. Наведіть загальну форму оператора для перехоплення всіх винятків при побудові системи обробки виняткових ситуацій для програми на C#.
13. Поясніть переваги та недоліки використання оператора для перехоплення всіх винятків при побудові системи обробки виняткових ситуацій для програми на C#.
14. Поясніть необхідність використання вкладених блоків try/catch при побудові системи обробки виняткових ситуацій для програми на C#.
15. Перерахуйте способи генерування винятків при побудові системи обробки виняткових ситуацій для програми на C#.
16. Поясніть необхідність створення винятку вручну при побудові системи обробки виняткових ситуацій для програми на C#.
17. Поясніть необхідність повторного генерування винятку при побудові системи обробки виняткових ситуацій для програми на C#.
18. Наведіть загальну форму оператора повторного генерування винятку при побудові системи обробки виняткових ситуацій для програми на C#.
19. Поясніть необхідність використання оператора finally при побудові системи обробки виняткових ситуацій для програми на C#.
20. Наведіть загальну форму використання блоків try/catch/finally при побудові системи обробки виняткових ситуацій для програми на C#.

ЗАВДАННЯ №11. СТАНДАРТНІ ТА КОРИСТУВАЦЬКІ ВИНЯТКИ

1. Поясніть призначення вхідного параметра оператора catch при побудові системи обробки виняткових ситуацій для програми на C#.
2. Назвіть властивість класу Exception, за допомогою якої можна отримати символічну строчку, що описує виняток для програми на C#.
3. Назвіть властивість класу Exception, за допомогою якої можна отримати символічну строчку із викликами стека, які призвели до винятку для програми на C#.
4. Назвіть властивість класу Exception, за допомогою якої можна отримати символічну строчку з назвою методу, який згенерував виняток для програми на C#.
5. Наведіть загальну форму кількох конструкторів класу Exception для програми на C#.
6. Назвіть клас, що відповідає винятку, що виникає через збереження значення у масиву при несумісності типів даних для програми на C#.
7. Назвіть клас, що відповідає винятку, що виникає через ділення на нуль для програми на C#.
8. Назвіть клас, який відповідає винятку, що виникає через звернення до індексу за межами масиву для програми на C#.
9. Назвіть клас, який відповідає винятку, що виникає через неправильне виконання динамічного приведення типів для програми на C#.
10. Назвіть клас, який відповідає винятку, що виникає через недостатність вільної пам'яті для програми на C#.
11. Назвіть клас, який відповідає винятку, що виникає через переповнення в результаті виконання арифметичної операції для програми на C#.
12. Назвіть клас, який відповідає винятку, що виникає через використання порожнього посилання для програми на C#.
13. Поясніть необхідність створення винятку користувача для програми на C#.
14. Назвіть базовий клас для класу винятку користувача для програми на C#.
15. Назвіть оператор, за допомогою якого виконується генерування винятку користувача для програми на C#.
16. Поясніть порядок розташування оператора для перехоплення всіх винятків та оператора для перехоплення винятку через ділення на нуль при побудові системи обробки виняткових ситуацій для програми на C#.
17. Поясніть порядок розташування операторів обробки винятків, вхідні параметри яких відносяться до базового та похідного типу винятків, при побудові системи обробки виняткових ситуацій для програми на C#.
18. Поясніть недолік механізму переповнення в результаті виконання арифметичної операції для програми на C#.
19. Наведіть загальну форму для спеціальних засобів, пов'язаних із генеруванням винятків через переповнення в результаті виконання арифметичної операції для програми на C#.
20. Поясніть функціонування спеціальних засобів, пов'язаних із генеруванням винятків через переповнення в результаті виконання арифметичної операції для програми на C#.

ЗАВДАННЯ №12. ДЕЛЕГАТИ ТА АНОНІМНІ ФУНКЦІЇ

1. Дайте визначення поняттю делегат для програми на C#.
2. Наведіть загальну форму оголошення делегата для програми на C#.
3. Поясніть особливості методу, на який може посилатися деякий делегат для програми на C#.
4. Поясніть механізм функціонування групового перетворення делегованих методів для програми на C#.
5. Перерахуйте види методів, які можуть бути призначені екземпляру делегата для програми на C#.
6. Поясніть механізм функціонування процесу групової адресації для делегатів для програми на C#.
7. Поясніть механізм функціонування процесу контраваріантності делегатів для програми на C#.
8. Поясніть механізм функціонування процесу коваріантності делегатів для програми на C#.
9. Назвіть базовий клас для делегатів для програми на C#.
10. Поясніть необхідність використання делегатів для програми на C#.
11. Дайте визначення поняттю анонімна функція для програми на C#.
12. Наведіть загальну форму оголошення анонімного методу для програми на C#.
13. Поясніть механізм визначення типів даних для вхідних параметрів в анонімному методі, який призначається екземпляру делегата для програми на C#.
14. Назвіть оператор, за допомогою якого повертається деяке значення з анонімного методу для програми на C#.
15. Дайте визначення поняттю зовнішня змінна відносно анонімного методу для програми на C#.
16. Дайте визначення поняттю захоплена змінна відносно анонімного методу для програми на C#.
17. Дайте визначення поняттю лямбда-вираз для програми на C#.
18. Назвіть лямбда-оператор, який використовується в лямбда-виразах для програми на C#.
19. Наведіть загальну форму оголошення одиночного лямбда-виразу для програми на C#.
20. Наведіть загальну форму оголошення блочного лямбда-виразу для програми на C#.

ЗАВДАННЯ №13. ОБРОБКА ПОДІЙ

1. Дайте визначення поняттю подія для програми на C#.

2. Поясніть механізм функціонування події для програми на C#.
3. Наведіть загальну форму оголошення події для програми на C#.
4. Назвіть тип члена класу, до якого належить подія для програми на C#.
5. Поясніть наслідки запуску події у випадку відсутності обробників подій для програми на C#.
6. Поясніть механізм функціонування процесу групової адресації події для програми на C#.
7. Назвіть оператори, які використовуються для реалізації групової адресації події для програми на C#.
8. Перерахуйте види методів, які можуть бути використані для реалізації групової адресації події для програми на C#.
9. Поясніть відмінність функціонування статичних методів та методів екземпляра класу, які призначаються як обробники події для програми на C#.
10. Поясніть необхідність використання події з аксесорами для програми на C#.
11. Наведіть загальну форму оголошення події з аксесорами для програми на C#.
12. Поясніть процес функціонування події з аксесорами для програми на C#.
13. Поясніть перевагу інтерфейсу, який містить подію для програми на C#.
14. Поясніть можливість абстрактного класу містити подію з аксесорами для програми на C#.
15. Поясніть необхідність використання лямбда-виразу в якості обробника події для програми на C#.
16. Поясніть недолік використання анонімного методу в якості обробника події для програми на C#.
17. Перерахуйте рекомендації, яких слід дотримуватися при оголошенні обробника події для сумісності програмних компонентів із середовищем .NET Framework для програми на C#.
18. Поясніть перевагу використання стандартного делегата .NET Framework при оголошенні події для програми на C#.
19. Поясніть можливість додавання нескінченної кількості обробників для події, що використовує стандартний делегат .NET Framework для програми на C#.
20. Поясніть можливість додавання обмеженої кількості обробників для події, що використовує стандартний делегат .NET Framework для програми на C#.

ЗАВДАННЯ №14. УЗАГАЛЬНЕНІ КЛАСИ

1. Поясніть призначення узагальненого класу для програми на C#.
2. Поясніть перевагу використання узагальненого класу для програми на C#.

3. Наведіть загальну форму оголошення узагальненого класу з одним параметром типу для програми на C#.
4. Наведіть загальну форму створення екземпляра узагальненого класу з одним параметром типу для програми на C#.
5. Поясніть належність екземплярів узагальненого класу з одним параметром типу, який приймає різні аргументи, до одного конкретного типу даних для програми на C#.
6. Поясніть принцип підвищення типової безпеки за допомогою узагальненого класу для програми на C#.
7. Наведіть загальну форму оголошення узагальненого класу із двома параметрами типу для програми на C#.
8. Наведіть загальну форму створення екземпляра узагальненого класу із двома параметрами типу для програми на C#.
9. Поясніть необхідність використання обмежень параметрів типу узагальнених класів для програми на C#.
10. Наведіть загальну форму оголошення узагальненого класу з обмеженням на базовий клас для параметра типу для програми на C#.
11. Поясніть перевагу використання узагальненого класу з обмеженням на базовий клас для параметра типу для програми на C#.
12. Наведіть загальну форму оголошення узагальненого класу з обмеженням на інтерфейс для параметра типу для програми на C#.
13. Поясніть перевагу використання узагальненого класу з обмеженням на інтерфейс для параметра типу для програми на C#.
14. Наведіть загальну форму оголошення узагальненого класу з обмеженням на конструктор для параметра типу для програми на C#.
15. Поясніть перевагу використання узагальненого класу з обмеженням на конструктор для параметра типу для програми на C#.
16. Наведіть загальну форму оголошення узагальненого класу з обмеженням типу посилання (типу даних значення) для параметра типу для програми на C#.
17. Поясніть перевагу використання узагальненого класу з обмеженням типу посилання (типу даних значення) для параметра типу для програми на C#.
18. Поясніть перевагу використання узагальненого класу з обмеженням у вигляді встановленого зв'язку для параметрів типу для програми на C#.
19. Наведіть правила призначення декількох обмежень для параметра типу узагальненого класу для програми на C#.
20. Поясніть необхідність використання оператора default для ініціалізації члена даних узагальненого класу параметра типу для програми на C#.

ЗАВДАННЯ №15. УЗАГАЛЬНЕНІ ТИПИ

1. Поясніть необхідність використання узагальненої структури для програми на C#.
2. Поясніть необхідність використання узагальненого методу для програми на C#.

3. Наведіть загальну форму виклику статичного узагальненого методу для програми на C#.
4. Наведіть загальну форму оголошення узагальненого методу з обмеженням для параметру типу для програми на C#.
5. Поясніть необхідність використання узагальненого делегата для програми на C#.
6. Поясніть особливості оголошення класу, який реалізує узагальнений інтерфейс для програми на C#.
7. Поясніть необхідність для узагальненого методу накладання для параметра типу обмеження на інтерфейс IEquatable для програми на C#.
8. Поясніть необхідність для узагальненого методу накладання для параметра типу обмеження на інтерфейс IComparable<T> для програми на C#.
9. Поясніть різницю між ієрархіями неузагальнених та узагальнених класів для програми на C#.
10. Поясніть особливості побудови ієрархії класів, де базовий клас є узагальненим для програми на C#.
11. Поясніть особливості побудови ієрархії класів, де похідний клас є узагальненим для програми на C#.
12. Поясніть особливості перевизначення віртуального узагальненого методу для програми на C#.
13. Поясніть особливості перевантаження узагальненого методу з кількома параметрами типу для програми на C#.
14. Поясніть необхідність використання коваріантності та контраваріантності для узагальненого інтерфейсу для програми на C#.
15. Назвіть обмеження, що накладаються на узагальнений інтерфейс, для якого встановлено коваріантність для програми на C#.
16. Назвіть обмеження, що накладаються на узагальнений інтерфейс, для якого встановлено контраваріантність для програми на C#.
17. Поясніть необхідність використання коваріантності та контраваріантності для узагальненого делегата для програми на C#.
18. Поясніть процес створення екземпляра узагальненого типу даних під час компіляції програми на C#.
19. Перерахуйте члени класу, які можуть бути узагальненими у неузагальненому класі для програми на C#.
20. Перерахуйте члени класу, які не можуть бути узагальненими у неузагальненому класі для програми на C#.

ЗАВДАННЯ №16. МОВА ІНТЕГРОВАНИХ ЗАПИТІВ

1. Поясніть призначення LINQ для програми на C#.
2. Поясніть перевагу використання LINQ для програми на C#.

3. Дайте визначення поняттю запит LINQ для програми на C#.
4. Назвіть інтерфейс, який має бути реалізований джерелом даних для запиту LINQ для програми на C#.
5. Назвіть тип даних змінної запиту LINQ для програми на C#.
6. Поясніть призначення трьох ключових слів для оголошення запиту LINQ для програми на C#.
7. Поясніть перевагу можливості неодноразового виконання запиту LINQ для програми на C#.
8. Поясніть механізм функціонування неодноразового виконання запиту LINQ для програми на C#.
9. Поясніть механізм визначення типу даних змінної діапазону запиту LINQ для програми на C#.
10. Перерахуйте ключові слова, які можна використовувати у запиті LINQ для програми на C#.
11. Перерахуйте ключові слова, якими може закінчуватися запит LINQ для програми на C#.
12. Поясніть особливості формування виразу оператора where запиту LINQ для програми на C#.
13. Поясніть особливості використання оператора orderby запиту LINQ для програми на C#.
14. Поясніть особливості використання оператора select запиту LINQ для програми на C#.
15. Поясніть особливості використання оператора from запиту LINQ для програми на C#.
16. Поясніть особливості використання вкладеного оператора from запиту LINQ для програми на C#.
17. Поясніть особливості використання оператора group запиту LINQ для програми на C#.
18. Назвіть тип даних змінної запиту LINQ, який закінчується оператором group для програми на C#.
19. Поясніть особливості використання оператора into запиту LINQ для програми на C#.
20. Перерахуйте оператори, після яких допустиме використання оператора into запиту LINQ для програми на C#.

ЗАВДАННЯ №17. НЕБЕЗПЕЧНИЙ КОД ТА ПОКАЖЧИКИ

1. Дайте визначення поняттю небезпечний код в C#.

2. Назвіть і дайте пояснення функціонуванню двох основних операторів, що застосовуються в покажчиках C#.
3. Назвіть ключове слово, яким повинен бути позначений код, у якому використовуються покажчики C#.
4. Поясніть необхідність використання модифікатора `fixed` у роботі з покажчиками в C#.
5. Перерахуйте можливі типи даних, які використовуються для оголошення покажчика в C#.
6. Перерахуйте арифметичні операції, які можна виконувати над покажчиками в C#.
7. Перерахуйте операції відношення, які можна виконувати над покажчиками в C#.
8. Поясніть особливості, що виникають під час роботи з масивом через покажчик на його початок в C#.
9. Поясніть особливості, що виникають під час роботи зі строчкою через покажчик на її початок в C#.
10. Дайте визначення поняттю багаторівнева непряма адресація в C#.
11. Назвіть та поясніть необхідність використання оператора, який виконує розподіл пам'яті, виділеної під стек в C#.
12. Поясніть необхідність використання буфера фіксованого розміру в C#.
13. Дайте визначення поняттю тип даних, що обнуляється в C#.
14. Наведіть приклад коду на C# оголошення цілочисельної змінної, що обнуляється, двома способами.
15. Поясніть процедуру перетворення типів даних між типами даних, що обнуляються і необнуляються, в математичних виразах C#.
16. Поясніть необхідність використання оператора `??` в C#.
17. Поясніть особливості, що виникають при використанні змінних типів даних, що обнуляються, у виразах відношення в C#.
18. Назвіть типи, які можуть бути частковими в C#.
19. Поясніть, які обмеження накладаються при оголошенні часткового методу в C#.
20. Поясніть відмінність змінних оголошених як `object` і `dynamic` в C#.

Навчальне видання

Методичні вказівки до виконання розрахунково-графічної роботи з освітнього компонента (навчальної дисципліни) Об'єктно-орієнтоване програмування для здобувачів першого

(бакалаврського) рівня вищої освіти за спеціальністю 125 Кібербезпека та захист інформації (частина 2) / Укл.: К.О. Трифонова. – Одеса, 2025. – 15 с.

Укладач: Трифонова К.О., ст. викл.

Підписано до друку _____. Формат 60x84/16. Папір газетний. Друк офсетний. 0,87
ум. друк. арк. 0,94 обл. - вид. арк.
Тираж 100 пр. Зам. №

Одеський національний морський університет
65029, Одеса, вул. Мечникова, 34