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

МЕТОДИЧНІ ВКАЗІВКИ

до виконання практичних робіт
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 практичних робіт
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 практичних робіт 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. – 16 с.

Укладач: Трифонова К.О., ст. викл.

ЗМІСТ

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.

ПРАКТИЧНА РОБОТА №9. ІНТЕРФЕЙСИ, СТРУКТУРИ І ПЕРЕРАХУВАННЯ

Мета роботи

1. Закріплення і поглиблення теоретичних знань, отриманих при вивченні розділу «Інтерфейси, структури і перерахування».
2. Набуття практичних навичок розробки та реалізації застосування з використанням інтерфейсів, структур і перерахувань.
3. Набуття практичних навичок налагодження застосування, в якому реалізовані інтерфейси, структури і перерахування.

Теоретичні відомості

Для виконання лабораторної роботи необхідні теоретичні відомості наводяться в конспекті лекцій у розділі «Інтерфейси, структури і перерахування».

Хід роботи

1. Назвіть основну перевагу використання інтерфейсу для програми на C#.
2. Назвіть обов'язок класу, який реалізує інтерфейс для програми на C#.
3. Назвіть кількість інтерфейсів, які можуть бути реалізовані в одному класі для програми на C#.
4. Поясніть неможливість наявності конструкторів в інтерфейсі для програми на C#.
5. Поясніть неможливість наявності статичних членів в інтерфейсі для програми на C#.
6. Наведіть загальну форму реалізації інтерфейсу в класі для програми на C#.
7. Наведіть приклад коду на C# оголошення та ініціалізації змінної посилального інтерфейсного типу.
8. Наведіть приклад коду на C# оголошення інтерфейсу з властивістю доступною лише для читання.
9. Наведіть приклад коду на C# оголошення інтерфейсу з індексатором, доступним лише для запису.
10. Наведіть приклад коду на C# оголошення ієрархії двох інтерфейсів та класу, що реалізує похідний інтерфейс.
11. Наведіть приклад коду на C# приховування імен при успадкуванні інтерфейсів.
12. Наведіть приклад коду на C# оголошення двох інтерфейсів з методами однакової сигнатури та класу, який реалізує обидва інтерфейси.
13. Наведіть приклад задачі, для вирішення якої доцільне застосування інтерфейсу замість абстрактного класу для програми на C#.
14. Наведіть приклад коду на C# використання деякого стандартного інтерфейсу, визначеного для середовища .NET Framework.
15. Наведіть приклад коду на C# оголошення структури та екземпляра структури.
16. Поясніть, у чому головна відмінність при виконанні процедури присвоювання екземплярів класів від екземплярів структур для програми на C#.
17. Наведіть приклад коду на C# оголошення перерахування та використання його в циклічному операторі.
18. Наведіть приклад коду на C# оголошення перерахування та ініціалізації його значень.
19. Наведіть приклад коду на C# оголошення перерахування із зазначенням деякого цілочисельного типу даних.
20. Наведіть приклад коду на C# оголошення перерахування та використання його в операторі вибору.

ПРАКТИЧНА РОБОТА №10. СИСТЕМА ОБРОБКИ ВИНЯТКОВИХ СИТУАЦІЙ

Мета роботи

1. Закріплення і поглиблення теоретичних знань, отриманих при вивченні розділу «Система обробки виняткових ситуацій».
2. Набуття практичних навичок розробки та реалізації застосування з використанням системи обробки виняткових ситуацій.
3. Набуття практичних навичок налагодження застосування, в якому реалізовано систему обробки виняткових ситуацій.

Теоретичні відомості

Для виконання лабораторної роботи необхідні теоретичні відомості наводяться в конспекті лекцій у розділі «Система обробки виняткових ситуацій».

Хід роботи

1. Назвіть форму, в якій представлені всі винятки для програми на C#.
2. Назвіть простір імен, де зберігаються всі класи винятків для програми на C#.
3. Назвіть базовий клас для всіх класів винятків для програми на C#.
4. Перерахуйте чотири оператори, які становлять основу системи обробки виняткових ситуацій для програми на C#.
5. Наведіть приклад коду на C# реалізації блоків, розташованих у методі Main(), які утворюють систему обробки виняткових ситуацій, в результаті виконання яких, здійснюється реагування на виняток через ділення на нуль.
6. Наведіть приклад коду на C# реалізації блоків, розташованих у методі Main(), які утворюють систему обробки виняткових ситуацій, в результаті виконання яких, здійснюється реагування на виняток через звернення до індексу за межами масиву.
7. Наведіть приклад коду на C# реалізації блоків, розташованих у методі Main(), які утворюють систему обробки виняткових ситуацій, в результаті виконання яких, здійснюється реагування на виняток через звернення до індексу за межами масиву в методі, який викликається в Main().
8. Поясніть наслідки відсутності системи обробки виняткових ситуацій при виконанні ділення на нуль для програми на C#.
9. Поясніть наслідки наявності системи обробки виняткових ситуацій ділення на нуль, при зверненні до індексу за межами масиву для програми на C#.
10. Наведіть приклад коду на C# реалізації блоків, розташованих у методі Main(), які утворюють систему обробки виняткових ситуацій, в результаті виконання яких, здійснюється реагування на кілька однакових винятків.
11. Наведіть приклад коду на C# реалізації блоків, розташованих у методі Main(), які утворюють систему обробки виняткових ситуацій, в результаті виконання, яких здійснюється реагування на кілька різних винятків.
12. Наведіть приклад коду на C# реалізації блоків, розташованих у методі Main(), які утворюють систему обробки виняткових ситуацій, в результаті виконання яких, здійснюється реагування на будь-які винятки.
13. Поясніть порядок розташування оператора для перехоплення всіх винятків відносно інших операторів при побудові системи обробки виняткових ситуацій для програми на C#.
14. Наведіть приклад коду на C# реалізації блоків, розташованих у методі Main(), які утворюють систему обробки виняткових ситуацій, в результаті виконання яких, здійснюється реагування на виняток через ділення на нуль і через звернення до індексу за межами масиву, після якого програма завершує свою роботу.

15. Наведіть приклад задачі, для вирішення якої доцільно застосування генерації винятку вручну при побудові системи обробки виняткових ситуацій для програми на C#.

16. Наведіть приклад коду на C# реалізації блоків, розташованих у методі Main(), які утворюють систему обробки виняткових ситуацій, в результаті виконання яких, здійснюється реагування на виняток згенерований вручну.

17. Наведіть приклад задачі, для вирішення якої доцільно застосування повторного генерування винятку при побудові системи обробки виняткових ситуацій для програми на C#.

18. Наведіть приклад коду на C# реалізації блоків, розташованих у методі Main(), які утворюють систему обробки виняткових ситуацій, в результаті виконання яких, здійснюється повторне генерування винятку.

19. Наведіть приклад задачі, для вирішення якої доцільно застосування оператора finally при побудові системи обробки виняткових ситуацій для програми на C#.

20. Наведіть приклад коду на C# реалізації блоків, розташованих у методі Main(), які утворюють систему обробки виняткових ситуацій, в результаті виконання яких, здійснюється реалізація деякого блоку коду, незалежно від причини виходу з блоку try.

ПРАКТИЧНА РОБОТА №11. СТАНДАРТНІ ТА КОРИСТУВАЦЬКІ ВИНЯТКИ

Мета роботи

1. Закріплення і поглиблення теоретичних знань, отриманих при вивченні розділу «Стандартні та користувацькі винятки».
2. Набуття практичних навичок розробки та реалізації застосування з використанням стандартних та користувацьких винятків.
3. Набуття практичних навичок налагодження застосування, в якому реалізовано стандартні та користувацькі винятки.

Теоретичні відомості

Для виконання лабораторної роботи необхідні теоретичні відомості наводяться в конспекті лекцій у розділі «Стандартні та користувацькі винятки».

Хід роботи

1. Назвіть тип даних вхідного параметра оператора для перехоплення всіх винятків при побудові системи обробки виняткових ситуацій для програми на C#.

2. Наведіть приклад коду на C# реалізації блоків, розташованих у методі Main(), які утворюють систему обробки виняткових ситуацій, в результаті виконання яких здійснюється реагування на виняток та виведення на консоль символьної строчки, що описує характер винятку.

3. Наведіть приклад коду на C# реалізації блоків, розташованих у методі Main(), які утворюють систему обробки виняткових ситуацій, в результаті виконання яких здійснюється реагування на виняток та виведення на консоль символьної строчки з викликами стека, які призвели до винятку.

4. Наведіть приклад коду на C# реалізації блоків, розташованих у методі Main(), які утворюють систему обробки виняткових ситуацій, в результаті виконання яких здійснюється реагування на виняток та виведення на консоль символьної строчки з назвою методу, який згенерував виняток.

5. Наведіть приклад винятку, що породжує внутрішній виняток для програми на C#.

6. Наведіть приклад коду на C# реалізації блоків, розташованих у методі Main(), які утворюють систему обробки виняткових ситуацій, в результаті виконання яких здійснюється реагування на виняток через збереження значення у масиві при несумісності типів даних.

7. Наведіть приклад коду на C# реалізації блоків, розташованих у методі Main(), які утворюють систему обробки виняткових ситуацій, в результаті виконання яких здійснюється реагування на виняток через ділення на нуль.

8. Наведіть приклад коду на C# реалізації блоків, розташованих у методі Main(), які утворюють систему обробки виняткових ситуацій, в результаті виконання яких здійснюється реагування на виняток через звернення до індексу за межами масиву.

9. Наведіть приклад коду на C# реалізації блоків, розташованих у методі Main(), які утворюють систему обробки виняткових ситуацій, в результаті виконання яких здійснюється реагування на виняток через неправильне виконання динамічного приведення типів.

10. Наведіть приклад коду на C# реалізації блоків, розташованих у методі Main(), які утворюють систему обробки виняткових ситуацій, в результаті виконання яких здійснюється реагування на виняток через недостатність вільної пам'яті для подальшого виконання програми.

11. Наведіть приклад коду на C# реалізації блоків, розташованих у методі Main(), які утворюють систему обробки виняткових ситуацій, в результаті виконання яких здійснюється реагування на виняток через переповнення в результаті виконання арифметичної операції.

12. Наведіть приклад коду на C# реалізації блоків, розташованих у методі Main(), які утворюють систему обробки виняткових ситуацій, в результаті виконання яких здійснюється реагування на виняток через використання порожнього посилання.

13. Наведіть приклад задачі, для вирішення якої доцільно створення винятку користувача для програми на C#.

14. Назвіть тип даних вхідного параметра оператора для перехоплення винятку користувача при побудові системи обробки виняткових ситуацій для програми на C#.

15. Наведіть приклад коду на C# реалізації блоків, розташованих у методі Main(), які утворюють систему обробки виняткових ситуацій, в результаті виконання яких здійснюється реагування на виняток користувача.

16. Наведіть приклад коду на C# реалізації блоків, розташованих у методі Main(), які утворюють систему обробки виняткових ситуацій, внаслідок виконання яких здійснюється реагування на будь-які винятки і безпосередньо на виняток через ділення на нуль.

17. Наведіть приклад коду на C# реалізації блоків, розташованих у методі Main(), які утворюють систему обробки виняткових ситуацій, в результаті виконання яких здійснюється реагування на винятки, що відносяться до базового та похідного класів.

18. Наведіть приклад задачі, для вирішення якої доцільно застосування генерації винятку через переповнення в результаті виконання арифметичної операції для програми на C#.

19. Поясніть наслідки відсутності системи обробки виняткових ситуацій при переповненні в результаті виконання арифметичної операції для програми на C#.

20. Наведіть приклад коду на C# реалізації блоків, розташованих у методі Main(), які утворюють систему обробки виняткових ситуацій, в результаті виконання яких не здійснюється реагування на виняток переповнення в результаті виконання арифметичної операції.

ПРАКТИЧНА РОБОТА №12. ДЕЛЕГАТИ ТА АНОНІМНІ ФУНКЦІЇ

Мета роботи

1. Закріплення і поглиблення теоретичних знань, отриманих при вивченні розділу «Делегати та анонімні функції».

2. Набуття практичних навичок розробки та реалізації застосування з використанням делегатів та анонімних функцій.

3. Набуття практичних навичок налагодження застосування, в якому реалізовані делегати та анонімні функції.

Теоретичні відомості

Для виконання лабораторної роботи необхідні теоретичні відомості наводяться в конспекті лекцій у розділі «Делегати та анонімні функції».

Хід роботи

1. Поясніть головну перевагу використання делегата для програми на C#.

2. Наведіть загальну форму оголошення екземпляра делегата для програми на C#.

3. Наведіть приклад коду на C# оголошення делегата та створення його екземпляра, якому призначено метод із вхідними та вихідними параметрами.

4. Наведіть приклад коду на C# оголошення делегата та створення його екземпляра, якому призначено метод без явного виклику конструктора делегата.

5. Наведіть приклад коду на C# оголошення делегата та створення його екземпляра, якому призначено метод екземпляра деякого класу.

6. Наведіть приклад коду на C# оголошення делегата та створення його екземпляра, якому призначено ланцюжок статичних методів.

7. Наведіть приклад коду на C# оголошення делегата та створення його екземпляра, якому призначено метод, вхідний параметр якого не збігається з вхідним параметром делегата.

8. Наведіть приклад коду на C# оголошення делегата та створення його екземпляра, якому призначено метод, вихідний параметр якого не збігається з вихідним параметром делегата.

9. Перерахуйте кілька методів класу Delegate для програми на C#.

10. Наведіть приклад задачі, для вирішення якої доцільно застосування делегата для програми на C#.

11. Поясніть необхідність використання анонімної функції для програми на C#.

12. Наведіть приклад коду на C# оголошення делегата та створення його екземпляра, якому призначено анонімний метод без вхідних та вихідних параметрів.

13. Наведіть приклад коду на C# оголошення делегата та створення його екземпляра, якому призначено анонімний метод із вхідними параметрами та без вихідного параметра.

14. Наведіть приклад коду на C# оголошення делегата та створення його екземпляра, якому призначено анонімний метод без вхідних параметрів та з вихідним параметром.

15. Наведіть приклад коду на C# оголошення делегата та створення його екземпляра, якому призначено анонімний метод, що захопив зовнішню змінну.

16. Поясніть час життя захопленої змінної анонімним методом, який присвоєно екземпляру делегата для програми на C#.

17. Поясніть необхідність використання лямбда-виразу для програми на C#.

18. Перерахуйте види лямбда-виразів для програми на C#.

19. Наведіть приклад коду на C# оголошення делегата та створення його екземпляра, якому призначено одиночний лямбда-вираз.

20. Наведіть приклад коду на C# оголошення делегата та створення його екземпляра, якому призначено блочний лямбда-вираз.

ПРАКТИЧНА РОБОТА №13. ОБРОБКА ПОДІЙ

Мета роботи

1. Закріплення і поглиблення теоретичних знань, отриманих при вивченні розділу «Обробка подій».

2. Набуття практичних навичок розробки та реалізації застосування з використанням обробки подій.

3. Набуття практичних навичок налагодження застосування, в якому реалізовано обробку подій.

Теоретичні відомості

Для виконання лабораторної роботи необхідні теоретичні відомості наводяться в конспекті лекцій у розділі «Обробка подій».

Хід роботи

1. Поясніть основне призначення події для програми на C#.

2. Наведіть приклад задачі, для вирішення якої доцільно застосування події для програми на C#.

3. Наведіть приклад коду на C# оголошення класу, який містить подію та метод, що виконує запуск події.

4. Назвіть тип даних, який лежить в основі події для програми на C#.

5. Назвіть кількість обробників події, які можуть бути додані для програми на C#.

6. Поясніть сигнатуру методів, які можна використовувати для групової адресації події для програми на C#.

7. Поясніть можливість розбіжності сигнатури методів, які використовуються для групової адресації події, з відповідним типом делегата для програми на C#.

8. Наведіть приклад коду на C# оголошення класу, який містить подію та метод, що виконує запуск події; створення екземпляра побудованого класу та додавання для його події в якості обробника нестатичного методу.

9. Наведіть приклад коду на C# оголошення класу, який містить подію та метод, що виконує запуск події; створення екземпляра побудованого класу та додавання для його події в якості обробника статичного методу.

10. Наведіть приклад задачі, для вирішення якої доцільно застосування події з аксесорами для програми на C#.

11. Наведіть приклад коду на C# оголошення класу, який містить подію з аксесорами та метод, що виконує запуск події.

12. Назвіть кількість обробників події, які можуть бути додані події з аксесорами для програми на C#.

13. Наведіть приклад коду на C# оголошення класу, який реалізує інтерфейс, що містить подію.

14. Наведіть приклад коду на C# оголошення класу, який реалізує абстрактний клас, що містить подію.

15. Наведіть приклад коду на C# оголошення класу, який містить подію та метод, що виконує запуск події; створення екземпляра побудованого класу та додавання для його події в якості обробника лямбда-вираз.

16. Наведіть приклад коду на C# оголошення класу, який містить подію та метод, що виконує запуск події; створення екземпляра побудованого класу та додавання для його події в якості обробника анонімного методу.

17. Наведіть приклад коду на C# оголошення .NET сумісного обробника події.

18. Наведіть приклад коду на C# оголошення події, використовуючи стандартний делегат .NET Framework.

19. Наведіть приклад коду на C# оголошення події, використовуючи стандартний делегат .NET Framework, якому може бути призначена нескінченна кількість обробників.

20. Наведіть приклад коду на C# оголошення події, використовуючи стандартний делегат .NET Framework, якому може бути призначена обмежена кількість обробників.

ПРАКТИЧНА РОБОТА №14. УЗАГАЛЬНЕНІ КЛАСИ

Мета роботи

1. Закріплення і поглиблення теоретичних знань, отриманих при вивченні розділу «Узагальнені класи».

2. Набуття практичних навичок розробки та реалізації застосування з використанням узагальнених класів.

3. Набуття практичних навичок налагодження застосування, в якому реалізовані узагальнені класи.

Теоретичні відомості

Для виконання лабораторної роботи необхідні теоретичні відомості наводяться в конспекті лекцій у розділі «Узагальнені класи».

Хід роботи

1. Перерахуйте типи даних, які можуть бути узагальненими в C#.

2. Наведіть приклад коду на C# узагальненого класу з одним параметром.

3. Поясніть, до якого типу даних належить екземпляр узагальненого класу в C#.

4. Поясніть причину типової безпеки узагальнених типів даних в C#.

5. Наведіть приклад коду на C# узагальненого класу із двома параметрами.

6. Наведіть загальну форму оголошення екземпляра узагальненого класу в C#.

7. Перерахуйте види обмежень для узагальнених типів даних в C#.

8. Поясніть необхідність використання узагальненого класу з обмеженням на базовий клас в C#.

9. Наведіть приклад коду на C# узагальненого класу з обмеженням на базовий клас.

10. Поясніть необхідність використання узагальненого класу з обмеженням на інтерфейс в C#.

11. Наведіть приклад коду на C# узагальненого класу з обмеженням на інтерфейс.

12. Поясніть необхідність використання узагальненого класу з обмеженням на конструктор в C#.

13. Наведіть приклад коду на C# узагальненого класу з обмеженням на конструктор.

14. Поясніть необхідність використання узагальненого класу з обмеженням типу посилання або типу значення в C#.

15. Наведіть приклад коду на C# узагальненого класу з обмеженням типу посилання.

16. Наведіть приклад коду на C# узагальненого класу з обмеженням типу значення.

17. Наведіть приклад коду на C# екземпляру узагальненого класу з обмеженням типу значення.

18. Поясніть необхідність використання узагальненого класу з двома параметрами, між якими встановлено зв'язок за допомогою обмеження в C#.

19. Поясніть, які обмеження не можуть бути встановлені одночасно на параметр типу для узагальненого класу в C#.

20. Наведіть приклад коду на C# конструктора за замовчуванням узагальненого класу з одним параметром, в якому виконується ініціалізація параметра типу.

ПРАКТИЧНА РОБОТА №15. УЗАГАЛЬНЕНІ ТИПИ

Мета роботи

1. Закріплення і поглиблення теоретичних знань, отриманих при вивченні розділу «Узагальнені типи».

2. Набуття практичних навичок розробки та реалізації програми з використанням узагальнених типів.

3. Набуття практичних навичок налагодження програми, в якій реалізовані узагальнені типи.

Теоретичні відомості

Для виконання лабораторної роботи необхідні теоретичні відомості наводяться в конспекті лекцій у розділі «Узагальнені типи».

Хід роботи

1. Поясніть необхідність використання узагальненої структури в C#.
2. Наведіть приклад коду на C# узагальненої структури.
3. Поясніть необхідність використання узагальненого методу в C#.
4. Наведіть приклад коду на C# узагальненого методу.
5. Поясніть принцип функціонування процесу виведення типів для узагальнених методів в C#.
6. Перерахуйте обмеження, які можуть накладатися на параметр типу для узагальненого методу в C#.
7. Поясніть необхідність використання узагальненого делегата в C#.
8. Поясніть особливості оголошення класу, який реалізує узагальнений інтерфейс в C#.
9. Наведіть приклад коду на C# узагальненого методу з одним параметром типу, який перевіряє на рівність вхідні параметри, що є параметрами типу.
10. Наведіть приклад коду на C# ієрархії двох узагальнених класів.
11. Наведіть приклад коду на C# ієрархії двох класів, де базовий клас є узагальненим.
12. Наведіть приклад коду на C# ієрархії двох класів, де похідний клас є узагальненим.
13. Наведіть приклад коду на C# ієрархії двох узагальнених класів, у базовому класі визначено узагальнений віртуальний метод, а в похідному класі даний метод перевизначено.
14. Наведіть приклад коду на C# узагальненого класу з двома параметрами типу, в якому реалізовано узагальнений метод із двома параметрами типу та його перевантажений варіант.
15. Перерахуйте узагальнені типи даних в C#, для яких можна скористатися коваріантністю і контраваріантністю.
16. Наведіть приклад коду на C# оголошення узагальненого інтерфейсу, для якого встановлена коваріантність.
17. Наведіть приклад коду на C# оголошення узагальненого інтерфейсу, для якого встановлена контраваріантність.
18. Наведіть приклад коду на C# оголошення узагальненого делегата, для якого встановлена коваріантність та контраваріантність.
19. Поясніть процес використання екземпляра узагальненого типу даних після компіляції під час виконання програми в C#.
20. Перерахуйте члени класу, які не можуть бути узагальненими в C#.

ПРАКТИЧНА РОБОТА №16. МОВА ІНТЕГРОВАНИХ ЗАПИТІВ

Мета роботи

1. Закріплення і поглиблення теоретичних знань, отриманих при вивченні розділу «Мова інтегрованих запитів».
2. Набуття практичних навичок розробки та реалізації програми з використанням мови інтегрованих запитів.
3. Набуття практичних навичок налагодження програми, в якій реалізовані інтегровані запити.

Теоретичні відомості

Для виконання лабораторної роботи необхідні теоретичні відомості наводяться в конспекті лекцій у розділі «Мова інтегрованих запитів».

Хід роботи

1. Перерахуйте, які джерела даних можуть бути використані у запиті LINQ в C#.

2. Назвіть узагальнений інтерфейс, який має бути реалізований у джерелі даних, для можливості звернення до нього за запитом LINQ в C#.

3. Поясніть, яким чином можна отримати результати виконання запиту LINQ в C#.

4. Поясніть, до якого типу даних відноситься змінна, що містить екземпляр, що повертається за запитом LINQ в C#.

5. Перерахуйте ключові слова, якими можуть закінчуватися запити LINQ в C#.

6. Поясніть необхідність використання запиту LINQ, який містить оператор where.

7. Наведіть приклад коду на C# оголошення запиту LINQ, який містить оператор where.

8. Поясніть можливість використання запиту LINQ, який містить декілька операторів where.

9. Поясніть необхідність використання запиту LINQ, який містить оператор orderby.

10. Наведіть приклад коду на C# оголошення запиту LINQ, який містить оператор orderby.

11. Поясніть необхідність використання запиту LINQ, який містить оператор select.

12. Наведіть приклад коду на C# оголошення запиту LINQ, який містить оператор select.

13. Поясніть необхідність використання запиту LINQ, який містить вкладені оператори from.

14. Наведіть приклад коду на C# оголошення запиту LINQ, який містить вкладені оператори from.

15. Поясніть необхідність використання запиту LINQ, який містить оператор group.

16. Наведіть приклад коду на C# оголошення запиту LINQ, який містить оператор group.

17. Поясніть можливість використання запиту LINQ, який містить декілька операторів group.

18. Поясніть необхідність використання запиту LINQ, який містить оператор into.

19. Наведіть приклад коду на C# оголошення запиту LINQ, який містить оператор into.

20. Поясніть можливість використання запиту LINQ, який містить декілька операторів into.

ПРАКТИЧНА РОБОТА №17. НЕБЕЗПЕЧНИЙ КОД ТА ПОКАЖЧИКИ

Мета роботи

1. Закріплення і поглиблення теоретичних знань, отриманих при вивченні розділу «Небезпечний код».

2. Набуття практичних навичок розробки та реалізації програми з використанням небезпечного коду.

3. Набуття практичних навичок налагодження програми, в якій реалізовано небезпечний код.

Теоретичні відомості

Для виконання лабораторної роботи необхідні теоретичні відомості наводяться в конспекті лекцій у розділі «Небезпечний код».

Хід роботи

1. Назвіть параметр компілятора для компілювання некерованого коду в C#.

2. Поясніть причину небезпеки використання покажчиків в C#.

3. Наведіть приклади видів коду C#, які можуть бути позначені як небезпечні за допомогою спеціального ключового слова.

4. Поясніть правила оголошення та використання покажчиків у C#.

5. Наведіть приклад коду на C#, в якому оголошено покажчик, захищений від видалення засобами збірки сміття.

6. Наведіть приклад коду на C# оголошення покажчика на структуру.

7. Наведіть приклад доступу до членів структури за допомогою покажчика у C#.

8. Поясніть використання арифметичних операцій над покажчиками у C#.

9. Поясніть, чим відрізняється в C# функціонування операції інкрементування для змінної, яка зберігає цілочисельне значення і для змінної, яка є покажчиком на цілочисельне значення.

10. Поясніть необхідність використання операцій відношення для покажчиків в C#.

11. Наведіть приклад порівняння двох покажчиків у C#.

12. Наведіть приклад коду на C# оголошення покажчика на початок масиву.

13. Наведіть приклад коду на C# оголошення покажчика на початок строчки.

14. Поясніть поняття багаторівневої непрямої адресації у C#.

15. Наведіть приклад коду на C# оголошення покажчика на покажчик значення цілочисельної змінної.

16. Наведіть приклад коду на C# оголошення масиву покажчиків на цілочисельні значення.

17. Наведіть приклад коду на C# для обходу масиву покажчиків.

18. Наведіть приклад коду на C# оголошення буфера фіксованого розміру.

19. Поясніть різницю між використанням fixed-буфера та звичайного масиву у C#.

20. Поясніть, які обмеження накладаються на використання покажчиків у керованому середовищі C#.

Навчальне видання

Методичні вказівки до виконання практичних робіт 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. – 16 с.

Укладач: Трифонова К.О., ст. викл.

Підписано до друку _____. Формат 60х84/16. 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