КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ТОРГОВЕЛЬНО-ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
СИСТЕМА УПРАВЛІННЯ ЯКІСТЮ
Система забезпечення якості освітньої діяльності та якості вищої освіти
сертифікована на відповідність ДСТУ ISO 9001:2015 / ISO 9001:2015
Кафедра інженерії програмного забезпечення та кібербезпеки

СИЛАБУС

ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ/ OBJECT-ORIENTED PROGRAMMING

SYLLABUS

освітній ступінь	**бакалавр /** bachelor
галузь знань	**12 Інформаційні технології /** Information Technologie
спеціальність	**121 Інженерія програмного забезпечення /** Software Engineering **125 Кібербезпека** /Cybersecurity

гарант освітньої програми

завідувач кафедри О.В. Криворучко

голова методичної ради факультету О.А. Харченко

Київ 2020

Розповсюдження і тиражування без офіційного дозволу КНТЕУ заборонено

Автори: Бебешко Б. Т., ст. викл.,
Хорольська К. В., асист.,

Силабус розглянуто і затверджено на засіданні кафедри програмної інженерії та інформаційних технологій 27 червня 2020 протокол № 31.

СИЛАБУС

ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ/ OBJECT-ORIENTED PROGRAMMING

SYLLABUS

освітній ступінь	**бакалавр /** bachelor
галузь знань	**12 Інформаційні технології /** Information Technologie
спеціальність	**121 Інженерія програмного забезпечення /** Software Engineering **125 Кібербезпека** /Cybersecurity

1. **Викладач**:

1.1. **Лектор**: Бебешко Богдан Тарасович,

- *вчене звання та посада*: старший викладач кафедри інженерії програмного забезпечення та кібербезпеки;
- *педагогічний стаж* – 1 рік;
- *e-mail:* b.bebeshko@knute.edu.ua ; thismushroom@gmail.com ;
- *наукові інтереси: всі види програмування.*
- *стажування та підвищення кваліфікації*: Курс «Введення в програмування ігор на Unity» Microsoft Visual Academy 2016 р.; Курс «Основи комп'ютерної безпеки» Microsoft Visual Academy 2016 р.; Курс «CS50x: Introduction to Computer Science» HarvardX, Harvard University, 2013 р.

1.2. **Асистент**: Хорольська Карина Вікторівна

- *вчене звання та посада:* асистент кафедри інженерії програмного забезпечення та кібербезпеки
- *педагогічний стаж* – 2 роки;
- *контактиний телефон:* (067) 6427852
- *e-mail:* k.khorolska@knute.edu.ua
- *наукові інтереси:* комп'ютерна техніка, новітні технології, хмарні технології.
- *стажування та підвищення кваліфікації:* Вища школа педагогічної майстерності, Київський національний торговельно-економічний університет, 2020р.Міжнародне підвищення кваліфікації (вебінар) - ВИКОРИСТАННЯ В СУЧАСНІЙ ОНЛАЙН ОСВІТІ МОЖЛИВОСТЕЙ ХМАРНИХ СЕРВІСІВ НА ПРИКЛАДІ ПЛАТФОРМИ GOOGLE MEET, GOOGLE CLASSROOM 2020 р. (Люблін, Республіка Польща).

2. **Дисципліна**: «**Об'єктно-орієнтовне програмування**»,

- рік навчання: II;
- семестр навчання: 3-4;
- кількість кредитів: 12;
- кількість годин за I семестр: 1800 год.
  - лекційних: 28 год.
  - лабораторних: 56 год.
  - на самостійне опрацювання: 96 год.
- кількість годин за II семестр: 180 год.
  - лекційних: 34 год.
  - лабораторних: 68 год.
  - на самостійне опрацювання: 78 год.
- кількість аудиторних годин на тиждень:
  - лекційних: 2 год.
  - лабораторних: 4 год.

3. **Час та місце проведення:**
   - аудиторні заняття - відповідно до розкладу КНТЕУ з врахуванням специфіки дисципліни проведення останньої передбачено в аудиторіях: 505, 510, 510а, 514;
   - поза аудиторна робота - самостійна робота студента, результат виконання якої висвітлено засобами Office 365;
   - всі лабораторні завдання виконуються на основі інтерактивних методів навчання у електронному середовищі. Передбачається можливість проведення лабораторних та лекційних занять на базах підприємств-партнерів.
4. **Пререквізити та постреквізити навчальної дисципліни:**
   - **пререквізити:** дисципліна базується на знаннях та компетентностях, що набуває здобувач вищої освіти під час вивчення дисциплін «Економічна інформатика», «Іноземна мова за професійним спрямуванням», «Інформаційні системи і технології».
   - **постреквізити:** Дисципліна надає студентам необхідні знання та навички, які будуть корисні при проходженні виробничої практики, підготовці до випускного кваліфікаційного проекту та у подальшій професійній діяльності.
   - **програмні результати навчання:** Використовувати інструментальні засоби розробки клієнт-серверних застосувань, проектувати концептуальні, логічні та фізичні моделі баз даних, розробляти та оптимізувати запити до них, створювати розподілені бази даних, сховища та вітрини даних, бази знань, у тому числі на хмарних сервісах, із застосуванням мов веб-програмування. Володіти навичками управління життєвим циклом програмного забезпечення, продуктів і сервісів інформаційних технологій відповідно до вимог і обмежень замовника, вміти розробляти проектну документацію (техніко-економічне обґрунтування, технічне завдання, бізнес-план, угоду, договір, контракт).

5. **Характеристика дисципліни:**

5.1. <u>Призначення навчальної дисципліни:</u> Дисципліна «Об'єктно-орієнтоване програмування» є важливою складовою підготовки сучасних фахівців ІТ-сфери. Вона є багатогранною та досить широкою, але з її допомогою можна суттєво підвищити свій рівень знань.

5.2.<u>Мета вивчення дисципліни:</u> метою вивчення дисципліни Об'єктно-орієнтоване програмування» є формування у студентів теоретичних знань та набуття ними практичних навичок в галузі розробки та програмування мовою C#/.

5.3. <u>Задачі вивчення дисципліни (відповідно до ОП):</u> Основними завданнями вивчення дисципліни «Об'єктно-орієнтоване програмування» є формування у студентів компетентностей та програмних результатів навчання:

5.4. <u>Зміст навчальної дисципліни:</u> відповідає навчальній та робочій програмі, яка відповідає запитам стейкхолдерів.

5.5. <u>План вивчення дисципліни:</u>

<u>ТЕОРЕТИЧНИЙ БЛОК:</u>

Навчальна діяльність	**Робочий час студента (год.)**
1	2
ТЕМА 1. Основи об'єктно-орієнтованого аналізу та проектування програмного забезпечення ***План:*** 1. Еволюція технології розробки програмного забезпечення. 2. Структурний та об'єктно-орієнтований підходи. 3. Поняття об'єктно-орієнтованого аналізу предметного середовища. 4. Загальні принципи побудови об'єктної модель предметного середовища. 5. Поняття об'єктно-орієнтованого проектування програмного забезпечення. 6. Поняття класу та об'єкту (екземпляру класу). 7. Поля та методи класу. 8. Взаємовідносини між класами та об'єктами. 9. Поняття успадкування. Композиція та колекція об'єктів. ***Рекомендовані джерела:*** ***Основні:*** *1, 2, 3.* ***Додаткові:*** *4, 5, 8.* ***Інтернет-ресурси:*** *9, 14, 15.*	4
ТЕМА 2. Об'єктно-орієнтоване програмування. Сучасні інструментальні засоби візуального програмування ***План:*** 1. Середовища візуального програмування. 2. Основні компоненти графічного інтерфейсу користувача. 3. Використання бібліотеки візуальних компонентів (меню, панелі інструментів, шаблони діалогових вікон тощо). 4. Побудова та опис діаграм прецедентів. ***Рекомендовані джерела:*** ***Основні:*** *1, 2, 3.* ***Додаткові:*** *4, 6, 7.*	4

1	2
Інтернет-ресурси: *9, 13, 14, 15.* ***Додатково:*** Дистанційний безкоштовний курс «Основи програмування C#» - https://courses.prometheus.org.ua/courses/Microsoft/CS201/2016_T1/about	
ТЕМА 3. Основи програмування керованого подіями. ***План:*** 1. Архітектура програм, орієнтованих на події. 2. Обробники подій від миші, клавіатури, команд меню, елементів управління тощо. 3. Побудова та опис діаграм взаємодії та компонентів. ***Рекомендовані джерела:*** ***Основні:*** *1, 2, 3.* ***Додаткові:*** *4-6, 8.* ***Інтернет-ресурси:*** *9, 13, 15.*	4
ТЕМА 4. Розробка класів та об'єктів мовою C#. ***План:*** 1. Опис класу. 2. Конструктори, деструктори класів. 3. Створення екземпляру класу. 4. Масиви об'єктів. 5. Статичні, константні члени класів. ***Рекомендовані джерела:*** ***Основні:*** *1, 2, 3.* ***Додаткові:*** *5, 7.* ***Інтернет-ресурси:*** *9, 13 - 15.*	4
ТЕМА 5. Інкапсуляція. ***План:*** 1. Поняття інкапсуляції. 2. Директиви видимості класів C# за замовчанням. 3. Організація доступу до захищених полів класу. 4. Дружні функції та класи. ***Рекомендовані джерела:*** ***Основні:*** *1, 2, 3.* ***Додаткові:*** *5, 6, 8.* ***Інтернет-ресурси:*** *9, 13 - 15.*	4

1	2
ТЕМА 6. Спадковість, просте та множинне успадкування. ***План:*** 1. Поняття спадковості. 2. Опис класів-нащадків в C# при одиничній спадковості. 3. Директиви видимості при спадкуванні в C#. 4. Явний виклик конструкторів-предків. 5. Послідовність викликів конструкторів/деструкторів для ієрархії класів. 6. Множинна спадковість в C#. 7. Проблеми ромбовидної спадковості, віртуальні предки. ***Рекомендовані джерела:*** ***Основні:*** *1, 2, 3.* ***Додаткові:*** *4, 7.* ***Інтернет-ресурси:*** *9 - 15.*	4
ТЕМА 7. Покажчики на об'єкти. Передача об'єктів як параметрів функцій. ***План:*** 1. Операції динамічного виділення пам'яті new та delete мови C#. 2. Особливості передачі параметрів-об'єктів за значенням мови C#. 3. Використання покажчиків та посилань на об'єкти. 4. Використання конструктору копії об'єкту. 5. Повернення об'єкту в якості результату роботи функції. ***Рекомендовані джерела:*** ***Основні:*** *1, 2, 3.* ***Додаткові:*** *4, 5.* ***Інтернет-ресурси:*** *9, 14 - 15.*	4
ТЕМА 8. Перевантаження функцій. ***План:*** 1. Неоднозначність при перевантаженні 2. Сумісність типів параметрів. 3. Параметри-посилання. 4. Параметри за замовчанням. 5. Перевантаження методів класу та конструкторів. ***Рекомендовані джерела:*** ***Основні:*** *1, 2, 3.* ***Додаткові:*** *4, 7, 8.* ***Інтернет-ресурси:*** *14, 15.*	4

1	2
Тема 9. Перевантаження операцій. ***План:*** 1. Перевантаження бінарних та унарних операції. 2. Перевантаження операції індексування. 3. Перевантаження операції присвоєння. 4. Особливості присвоєння об'єктів мови C#. 5. Перевантаження операцій потокового уведення-виведення. ***Рекомендовані джерела:*** ***Основні:*** *1, 2, 3.* ***Додаткові:*** *6, 7.* ***Інтернет-ресурси:*** *14, 15.*	4
Тема 10. Поліморфізм. ***План:*** 1. Поняття раннього та пізнього зв'язування. 2. Поняття перекриття та заміщення методів. 3. Механізм виклику заміщуваних методів, таблиці віртуальних методів. 4. Використання покажчиків на класи, як спосіб застосування поліморфізму. 5. Виклик заміщуваних методів нащадків з класів предків. 6. Абстрактні методи. ***Рекомендовані джерела:*** ***Основні:*** *1, 2, 3.* ***Додаткові:*** *5, 7.* ***Інтернет-ресурси:*** *14, 15.*	4
Тема 11. Шаблони функцій і класів. ***План:*** 1. Поняття шаблону функції. 2. Опис шаблону. 3. Створення екземпляру шаблону функції. 4. Відмінності перевантаження функції та шаблонів функції. 5. Опис шаблону-класу. 6. Створення екземпляру шаблону класу. ***Рекомендовані джерела:*** ***Основні:*** *1, 2, 3.* ***Додаткові:*** *5, 7.* ***Інтернет-ресурси:*** *14, 15.*	4

1	2
Тема 12. Бібліотека стандартних шаблонів STL: основні поняття. ***План:*** 1. Складові бібліотеки STL: контейнери, ітератори, алгоритми. 2. Базові класи-контейнери, їх загальна характеристика. 3. Особливості роботи з елементами контейнера небазових типів. 4. Ітератори як покажчики на елементи контейнерів, типи ітераторів. 5. Загальне поняття алгоритму. ***Рекомендовані джерела:*** ***Основні:*** *1, 2, 3.* ***Додаткові:*** *4-6.* ***Інтернет-ресурси:*** *14, 15.*	
Тема 13. Використання класів-контейнерів бібліотеки STL: vector, list, map. ***План:*** 1. Загальна структура контейнерів. 2. Клас-контейнер vector. Способи створення. 3. Методи класу вектор. Ітератори на елементи вектора. 4. Клас-контейнер list. Способи створення. 5. Методи класу списка. Ітератори на елементи списка. 6. Клас-контейнер map. Способи створення. 7. Методи класу асоціативний список. Ітератори на елементи. 8. Послідовності. 9. Адаптери послідовностей 10. Асоціативні масиви. 11. Бітова множина ***Рекомендовані джерела:*** ***Основні:*** *1, 2, 3.* ***Додаткові:*** *5, 7.* ***Інтернет-ресурси:*** *14, 15.*	4
Тема 14. Застосування алгоритмів бібліотеки STL. ***План:*** 1. Базові алгоритми бібліотеки STL. 2. Поняття предикату, унарний та бінарний предикат. 3. Приклади використання алгоритмів. ***Рекомендовані джерела:*** ***Основні:*** *1, 2, 3.* ***Додаткові:*** *4, 7.*	4

1	2
Інтернет-ресурси: *14, 15.*	
Тема 15. Класи потоків уведення та виведення. ***План:*** 1. Перевантаження операцій потокового уведення/виведення. 2. Файлове уведення/виведення. 3. Файлові потоки. 4. Управління потоковим уведенням/виведенням. 5. Переадресація введення та виведення 6. Визначення потокових операцій як дружніх 7. Маніпулятори. ***Рекомендовані джерела:*** ***Основні:*** *1, 2, 3.* ***Додаткові:*** *4 - 6.* ***Інтернет-ресурси:*** *14, 15.*	2
Тема 16. Оброблення виняткових ситуацій в C#. ***План:*** 1. Використання оператору try-catch. 2. Повторне збудження виключної ситуації throw. 3. Властивість Message, короткий опис помилки. 4. Властивість TargetSite - об'єкт, в якому виникла помилка. 5. Властивість InnerException – повернення наступного об'єкту в ланцюжку винятків. ***Рекомендовані джерела:*** ***Основні:*** *1, 2, 3.* ***Додаткові:*** *7, 8.* ***Інтернет-ресурси:*** *14, 15.*	2
Тема 17. Розроблення DLL-бібліотек. ***План:*** 1. Базові алгоритми бібліотеки DLL. 2. Створення динамічної dll шляхом перекомпіляції в il. 3. Динамічна dll + WPF. 4. Приклади використання бібліотеки DLL. ***Рекомендовані джерела:*** ***Основні:*** *1, 2, 3.* ***Додаткові:*** *5, 6.* ***Інтернет-ресурси:*** *14, 15.*	2
Всього	62

ЛАБОРАТОРНІ ЗАНЯТТЯ:

Навчальна діяльність	Робочий час студента (год.)	Оцінювання (бал)
1	2	3
Лабораторна робота 1. ***Тема: Механізми проектування мовою UML.*** *Завдання.* Провести об'єктно-орієнтований аналіз та проектування мовою UML майбутнього програмного продукту. ***Рекомендовані джерела:*** ***Основні:*** *1, 2, 3.* ***Додаткові:*** *4, 5, 8.* ***Інтернет-ресурси:*** *9, 14, 15.*	8	10
Лабораторна робота 2. ***Тема: Інструментальні засоби візуального програмування*** Завдання: Мотивація створення ООП, альтернативні підходи, тонкощі використання. Реалізація інкапсуляції та принципу DRY з та без ООП в C#. ***Рекомендовані джерела:*** ***Основні:*** *1, 2, 3.* ***Додаткові:*** *4, 6, 7.* ***Інтернет-ресурси:*** *9, 13, 14, 15.*	8	10
Лабораторна робота 3. ***Тема: Визначення основ програмування керованого подіями.*** *Завдання:* Перетворити команди проміжкової мови Intermediate Language (IL) в команди процессора у відповідності з індивідуальним завданням. ***Рекомендовані джерела:*** ***Основні:*** *1, 2, 3.* ***Додаткові:*** *4-6, 8.* ***Інтернет-ресурси:*** *9, 13, 15*	8	10
Лабораторна робота 4. ***Тема: Робота з класами та об'єктами мовою C#.*** *Завдання:* Створіть який-небудь клас (наприклад клас Автомобіль), оголосіть в ньому поля з різними режимами доступу (private, protected, internal, public). Створіть об'єкт класу, і спробуйте записати / отримати дані з різних полів. Відпрацювати данне завдання на 5 класах, які Ви створите особисто.	8	10

1	2	3
Рекомендовані джерела: ***Основні:*** *1, 2, 3.* ***Додаткові:*** *5, 7.* ***Інтернет-ресурси:*** *9, 13 - 15.*		
Лабораторна робота 5. ***Тема: Реалізація інкапсуляції в C#.*** *Завдання.* Реалізувати інкапсуляцію в C#, використовуючи модифікатори доступу public, protected, internal, private, protected internal. ***Рекомендовані джерела:*** ***Основні:*** *1, 2, 3.* ***Додаткові:*** *5, 6, 8.* ***Інтернет-ресурси:*** *9, 13 - 15.*	8	10
Лабораторна робота 6. ***Тема: Спадковість, просте та множинне успадкування.*** *Завдання:* Створіть базовий клас Геометрична фігура, передбачте в ньому загальні поля / властивості, наприклад координати центру фігури, за допомогою конструктора повинна бути можливість задати центр. На базі цього класу створіть два нових - Трикутник і Окружність. У цих класах повинні бути свої особливі поля, наприклад радіус для кола. В обидва класи додайте метод Намалювати, в якому могла б бути специфічна логіка малювання фігури. Створіть об'єкти трикутник і коло. ***Рекомендовані джерела:*** ***Основні:*** *1, 2, 3.* ***Додаткові:*** *4, 7.* ***Інтернет-ресурси:*** *9 - 15.*	8	10
Лабораторна робота 7. ***Тема: Операції NEW ТА DELETE.*** *Завдання:* 1. В індивідуальному завданні об'єднати всі дії, необхідні для створення об'єкта, в одну операцію, яку виконує оператор new. 2. За допомогою оператора delete оптимізуйте команду, яка знищує динамічно створений об'єкт MyType і звільняє займану їм пам'ять. ***Рекомендовані джерела:*** ***Основні:*** *1, 2, 3.* ***Додаткові:*** *4, 5.* ***Інтернет-ресурси:*** *9, 14 - 15.*	8	10

1	2	3
Лабораторна робота 8 ***Тема: Перевантаження функцій.*** *Завдання.* скласти й реалізувати алгоритми та програми мовою C#, які використовують функції, створені користувачем. У відповідності з індивідуальним завданням. ***Рекомендовані джерела:*** ***Основні:*** *1, 2, 3.* ***Додаткові:*** *4, 7, 8.* ***Інтернет-ресурси:*** *14, 15.*	8	5
Лабораторна робота 9 ***Тема: Перевантаження операцій.*** *Завдання:* Перевизначити (перевантажити) звичні знаки операцій таким чином, щоб вони стосувались екземплярів класів. Наприклад, нехай визначений класс **complex**, що реалізує комплексне число. Також можна реалізувати дії з екземплярами цього класу як звертання до певних функцій класу, проте значно зручніше було б оперувати звичними для сприйняття виразами на кшталт: **complex a, b; complex c = a + b; c =a*b;** Завдання виконати у відповідності з індивідуальним завданням. ***Рекомендовані джерела:*** ***Основні:*** *1, 2, 3.* ***Додаткові:*** *6, 7.* ***Інтернет-ресурси:*** *14, 15.*	8	5
Лабораторна робота 10 ***Тема: Поліморфізм C#*** *Завдання:* Використовуючи можливості поліморфізму напишіть більш абстрактні, розширювані програми, один і той же код використовується для об'єктів різних класів, поліпшується читабельність коду. Поліморфізм має позбавити розробника від написання, читання і налагодження безлічі if-else / switch-case конструкцій. ***Рекомендовані джерела:*** ***Основні:*** *1, 2, 3.* ***Додаткові:*** *5, 7.* ***Інтернет-ресурси:*** *14, 15.*	8	5
Лабораторна робота 11 ***Тема: Шаблони функцій і класів.*** *Завдання*:	8	5

1	2	3
1.Реалізувати один і той самий алгоритм для кожного потрібного типу даних; -використати абстрактний базовий клас, в якому визначити даний алгоритм і заміщати його відповідним чином у похідних класах; -використати засоби препроцесора і створити макровизначення. 2. Оцінити вади кожного із цих шляхів. ***Рекомендовані джерела:*** ***Основні:*** *1, 2, 3.* ***Додаткові:*** *5-8.* ***Інтернет-ресурси:*** *14, 15.*		

1	2	3
Лабораторна робота 12 ***Тема: Оголошення типів класів контейнера*** *Завдання:* 1. У кожному контейнері визначити службові типи: value_type, allocator_type, size_type, iterator, reverse_iterator, reference, key_type, key_compare і mapped_type. 2. У кожному контейнері визначити члени, що повертають ітератори: begin(), end(), rbegin() і rend(). 3. У кожному контейнері здійснти доступ до елементів функціями-членами front(), back(), operator [](i) і at(i). 4. Включити елементи у контейнер шляхом функції-членів insert(p, x), insert(p, n, x), insert(p, first, last), push_back(x) і push_front(x). 5. Видалення елементів у контейнері функціями-членами. pop_back(), pop_front(), erase(p), erase(first, last) і clear(). 6. Визначити службові операції контейнера - size(), empty(), capacity(), reserve(n), resize(n), swap(x), а також ==, !=, <. ***Рекомендовані джерела:*** ***Основні:*** *1, 2, 3.* ***Додаткові:*** *4-6.* ***Інтернет-ресурси:*** *14, 15.*	8	10
Лабораторна робота 13 ***Тема: Використання класів-контейнерів бібліотеки STL: vector, list, map.*** *Завдання:* Відтворити загальну структуру контейнерів. Кожен контейнер повинен задовольняти наступним вимогам: 1. Контейнер може містити об'єкти будь-якого типу, крім посилань. 2. Конструктор копіювання об'єкта, що втримується в контейнері, повинен виконувати операцію вставки. 3. Виділення й звільнення пам'яті, зайнятої об'єктом, що зберігається в контейнері, повинне здійснюватися автоматично. 4. Контейнер повинен допускати автоматичну зміну розмірів. 5. Деструктор контейнера повинен викликати деструктори об'єктів, що зберігаються в ньому.	8	10

1	2	3
!!! Якщо конструктор містить вказівники, звільнення пам'яті здійснюється окремою операцією. Для стандартизації імен типів елементів, що зберігаються в контейнері, у контейнерних класах утримується оператор typedef, що уніфікує імена. ***Рекомендовані джерела:*** ***Основні:*** *1, 2, 3.* ***Додаткові:*** *5, 7.* ***Інтернет-ресурси:*** *14, 15.*		
Лабораторна робота 14 ***Тема: Застосування алгоритмів бібліотеки STL.*** *Завдання:* <u>Перший етап</u> виконання роботи полягає у побудові програми розв'язання задачі згідно вказаних варіантів з використанням шаблонів-класів <vector> (клас масиву) , <algorithm> (клас алгоритмів) та ітератора бібліотеки STL. Використання шаблонів-класів здійснюється шляхом включення відповідних заголовочних файлів. Слід також використати основні методи контейнерних класів, наприклад begin()- повернення ітератора до початкового елементу масиву контейнера, end()- повернення ітератора на позицію останнього елементу контейнера, size()-повернення кількості елементів масиву контейнера. <u>Другий етап</u> - побудувати програми розв'язання задачі згідно вказаних варіантів з використанням власного шаблону-класу <VECTOR> (клас контейнера масиву) та дружнього класу ітератора. Контейнерний клас повинен містити дані масиву та основні методи, що були вище обумовлені, як основні методи контейнерних класів. Клас ітераторів повинен мітити основні функції роботи ітераторів, що також обумовленні вище. При виконанні завдання звертайте увагу на вищенаведені приклади. ***Рекомендовані джерела:*** ***Основні:*** *1, 2, 3.* ***Додаткові:*** *4, 7.* ***Інтернет-ресурси:*** *14, 15.*	8	10

1	2	3
Лабораторна робота 15 ***Тема: Класи потоків уведення та виведення.*** *Завдання:* Відпрацювати переадресацію введення та виведення перевизначивши імена cin або cout власним об'єктам-потокам. Таке призначення дозволить програмі легко переадресувати стандартні операції введення або виведення. ***Рекомендовані джерела:*** ***Основні:*** *1, 2, 3.* ***Додаткові:*** *4 - 6.* ***Інтернет-ресурси:*** *14, 15.*	8	10
Лабораторна робота 16 ***Тема: Оброблення виняткових ситуацій в C#.*** *Завдання:* В програмному коді при обробці помилок застосувати спрощену конструкція try/catch. Також віпрацювати другіи спосіб - використати конструкцію try/catch/finally. ***Рекомендовані джерела:*** ***Основні:*** *1, 2, 3.* ***Додаткові:*** *7, 8.* ***Інтернет-ресурси:*** *14, 15.*	8	5
Лабораторна робота 17 ***Тема. Створення та зв'язування динамічних бібліотек DLL.*** Завдання: 1. Створити нову DLL з ім'ям MyDLL. 2. Створити заголовний файл бібліотеки 3. Виконайте команду Project | Options і у вікні опцій проекту на сторінці Linker переконайтеся, що включено індикатор опції Generate import library.. 4. Виконайте команду Project | Build MyDLL. У результаті будуть створені файли MyDLL.dll і MyDLL.lib. Перевірте їхню наявність у робочому каталозі. 5. Виконайте команду View | Project Manager. 6. Розмістіть на формі проекту вікно редагування й кнопку. 7. Включіть у модуль після директиви препроцесора #pragma hdrstop директиву, що підключає заголовний файл бібліотеки: 8. Виконайте даний додаток і переконайтеся, що все	8	5

1	2	3
працює нормально. У випадку виникнення, виправте всі помилки в покроковому режимі. 9. Необхідно реалізувати динамічне зв'язування DLL бібліотеки. 10. Розмістіть на формі вікно редагування й три кнопки. Кнопка Завантажити (ім'я BLoad) буде забезпечувати завантаження DLL, кнопка Вивантажити (ім'я BFree) буде вивантажувати бібліотеку з пам'яті, а кнопка Кодувати/Декодувати (ім'я Button1) аналогічна кнопці при статичному зв'язуванні 11. У заголовному файлі додатка в опис класу форми внесіть наступні оголошення: 12.Пропишіть коди оброблювачів подій для кнопок додатка 13. Виконайте даний додаток і переконайтеся, що все працює нормально. При необхідності виправити помилки ***Рекомендовані джерела:*** ***Основні:*** *1, 2, 3.* ***Додаткові:*** *5, 6.* ***Інтернет-ресурси:*** *14, 15.*		

** всі лабораторні завдання виконуються на основі інтерактивних методів навчання у комп'ютерному середовищі*

<u>САМОСТІЙНА РОБОТА:</u>

Навчальна діяльність	**Робочий час студента (год.)**	**Оцінювання (бал)**
1	2	3
Тема 1. Основи об'єктно-орієнтованого аналізу та проектування програмного забезпечення ***Рекомендовані джерела:*** ***Основні:*** *1, 2, 3.* ***Додаткові:*** *4, 5, 8.* ***Інтернет-ресурси:*** *9, 14, 15.* **Завдання для самостійної роботи:** Вивчення та доповнення матеріалів лекції на основі самостійного опрацювання основних літературних джерел, зазначених у списку та електронних	14	3

1	2	3
матеріалів, які надаються викладачем. **Самостійна робота студентів.** Вивчення матеріалу лекції. Підготувати доповідь у вигляді презентації на одну з вказаних тем: 1. Визначення об’єктно-орієнтованого аналізу, проектування та програмування. 2. Основи об’єктно-орієнтованого проектування програмного забезпечення мовою UML. 3. Основні діаграми UML. 4. Правила побудови UML-діаграми класів		
Тема 2. Об’єктно-орієнтоване програмування. Сучасні інструментальні засоби візуального програмування ***Рекомендовані джерела:*** ***Основні:*** *1, 2, 3.* ***Додаткові:*** *4, 6, 7.* ***Інтернет-ресурси:*** *9, 13, 14, 15.* **Завдання для самостійної роботи:** Вивчення та доповнення матеріалів лекції на основі самостійного опрацювання основних літературних джерел, зазначених у списку та електронних матеріалів, які надаються викладачем. **Самостійна робота студентів.** Вивчення матеріалу лекції, підготовка до лабораторного заняття. Опанувати дистанційно безкоштовний курс «Основи програмування C#». Цей курс може бути корисний для всіх початківців розробників на мові програмування C#, оскільки матеріал містить всі базові поняття, припускаючи, що слухач не мав досвіду розробки раніше. В ході курсу, слухач дізнається про основні концепції мови, починаючи з загальної системи типів і, закінчуючи, об'єктно-орієнтованим програмуванням. Окремі модулі курсу будуть присвячені найбільш корисним класами в .NET Framework, які знадобляться при подальшому вивченні таких питань, як розробка для Windows 10. ***https://courses.prometheus.org.ua/courses/Microsoft/CS201/2016_T1/about***	14	3

1	2	3
Тема 3. Основи програмування керованого подіями. ***Рекомендовані джерела:*** ***Основні:*** *1, 2, 3.* ***Додаткові:*** *4-6, 8.* ***Інтернет-ресурси:*** *9, 13, 15* **Завдання для самостійної роботи:** Вивчення матеріалів лекції та самостійне опрацювання основних літературних джерел, зазначених у списку та електронних матеріалів, які надаються викладачем. Підготувати доповідь у вигляді презентації на одну з вказаних тем: 1. Що собою являє технологія .Net Framework? 2. За рахунок чого досягається міжмовна інтеграція у .Net Framework? 3. За рахунок чого досягається незалежність від платформи у .Net Framework? 4. Які основні можливості надає середовище Common Language Runtime? 5. Яка мова програмування є основною мовою програмування технології .Net Framework?	14	3
Тема 4. Розробка класів та об'єктів мовою C#. ***Рекомендовані джерела:*** ***Основні:*** *1, 2, 3.* ***Додаткові:*** *5, 7.* ***Інтернет-ресурси:*** *9, 13 - 15.* **Завдання для самостійної роботи:** Вивчення матеріалів лекції та самостійне опрацювання основних літературних джерел, зазначених у списку та електронних матеріалів, які надаються викладачем. Підготувати доповідь у вигляді презентації: Проведіть аналіз Констант та представте у вигляді презентації. Особливу увагу звернути на Константи - члени класу та Константи – змінні. Наведіть конкретні приклади у презентації оголошення константи.	14	3
Тема 5. Інкапсуляція. ***Рекомендовані джерела:*** ***Основні:*** *1, 2, 3.* ***Додаткові:*** *4-6.* ***Інтернет-ресурси:*** *14, 15.* **Завдання для самостійної роботи:**	14	3

1	2	3
Вивчення матеріалів лекції та самостійне опрацювання основних літературних джерел, зазначених у списку та електронних матеріалів, які надаються викладачем. Підготувати доповідь у вигляді презентації: Провести порівняльний аналіз модифікаторів доступу в C#, а саме public, protected, internal, private, protected internal. Порівняння підкріпити прикладами.		

1	2	3
Тема 6. Спадковість, просте та множинне успадкування. ***Рекомендовані джерела:*** ***Основні:*** *1, 2, 3.* ***Додаткові:*** *4, 7.* ***Інтернет-ресурси:*** *9 – 15* **Завдання для самостійної роботи:** Вивчення матеріалів лекції та самостійне опрацювання основних літературних джерел, зазначених у списку та електронних матеріалів, які надаються викладачем. Підготувати доповідь у вигляді презентації на одну з вказаних тем: 1. Доступ до членів базового класу з класу послідовника. 2. Доступ до членів базового класу, чкі оголошені як public, protected, internal ат protected internal. 3. Члени базового класу з модифікатором доступу privat.	8	5
Тема 7. Покажчики на об'єкти. Передача об'єктів як параметрів функцій ***Рекомендовані джерела:*** ***Основні:*** *1, 2, 3.* ***Додаткові:*** *4, 5.* ***Інтернет-ресурси:*** *9, 14 - 15.* **Завдання для самостійної роботи:** Вивчення матеріалів лекції та самостійне опрацювання основних літературних джерел, зазначених у списку та електронних матеріалів, які надаються викладачем. Підготувати доповідь у вигляді презентації на одну з вказаних тем: 1. Перелік бібліотечних файлів, які належать бібліотеці часу виконання C 2. Перелік бібліотечних файлів, які належать до стандартної библиотеки C# 3. Функції бібліотеки CRT.	12	5
Тема 8. Перевантаження функцій. ***Рекомендовані джерела:*** ***Основні:*** *1, 2, 3.* ***Додаткові:*** *4, 7, 8.* ***Інтернет-ресурси:*** *14, 15.* **Завдання для самостійної роботи:** Вивчення матеріалів лекції та самостійне	6	3

1	2	3
опрацювання основних літературних джерел, зазначених у списку та електронних матеріалів, які надаються викладачем. На власному індивідуальному завданні розкрити питання: 1. Перевантаження функцій - простий поліморфізм. 2. Поняття сигнатур.		

1	2	3
Тема 9. Перевантаження операцій. ***Рекомендовані джерела:*** ***Основні:*** *1, 2, 3.* ***Додаткові:*** *6, 7.* ***Інтернет-ресурси:*** *14, 15.* **Завдання для самостійної роботи:** Вивчення матеріалів лекції та самостійне опрацювання основних літературних джерел, зазначених у списку та електронних матеріалів, які надаються викладачем. Підготувати доповідь у вигляді презентації на одну з вказаних тем: 1. Операторні функції 2. Бінарні операції 3. Декларація функції-члену класу для перевантаження бінарної операції	6	3
Тема 10. Поліморфізм. ***Рекомендовані джерела:*** ***Основні:*** *1, 2, 3.* ***Додаткові:*** *5, 7.* ***Інтернет-ресурси:*** *14, 15.* **Завдання для самостійної роботи:** Вивчення матеріалів лекції та самостійне опрацювання основних літературних джерел, зазначених у списку та електронних матеріалів, які надаються викладачем. Підготувати доповідь у вигляді презентації на одну з вказаних тем: 1. Віртуальний метод – як, метод, який МОЖЕ бути перевизначений в класі-спадкоємця. 2. Абстрактний метод – як, метод, який повинен бути реалізований в класі-спадкоємця. 3. Перевизначення методу – як, зміна реалізації методу, встановленого як віртуальний.	6	3
Тема 11. Шаблони функцій і класів. ***Рекомендовані джерела:*** ***Основні:*** *1, 2, 3.* ***Додаткові:*** *5-8.* ***Інтернет-ресурси:*** *14, 15.* **Завдання для самостійної роботи:** Вивчення матеріалів лекції та самостійне опрацювання основних літературних джерел, зазначених у списку та електронних матеріалів, які надаються викладачем.	4	3

1	2	3
Розкрити тему у власних прикладах: Що відбувається при виклику функції, визначеної шаблоном? - перевантаження шаблонів може привести до складних і неоднозначних виборів функцій-кандидатів.		
Тема 12. Бібліотека стандартних шаблонів STL: основні поняття. ***Рекомендовані джерела:*** ***Основні:*** *1, 2, 3.* ***Додаткові:*** *4-6.* ***Інтернет-ресурси:*** *14, 15.* **Завдання для самостійної роботи:** Вивчення матеріалів лекції та самостійне опрацювання основних літературних джерел, зазначених у списку та електронних матеріалів, які надаються викладачем. *Використавуючи власний програмний продукт наведіть приклади по питанням, які наведені нижче:* 1. Що таке стандартні алгоритми? Наведіть приклад. 2. Що таке ітератори? Які операції можна виконувати над усіма типами літераторів? 3. Які класи належать до категорії контейнерів? 4. На які групи розподіляються алгоритми? 5. Які алгоритми належать до групи тих, що не модифікують елементи контейнера? 6. Які алгоритми належать до групи тих, що модифікують елементи контейнера? 7. Які алгоритми належать до групи тих, що виконують операції над множинами? 8. Які алгоритми належать до групи тих, що що знаходять мінімальні і максимальні елементи контейнера? 9. За допомогою яких операцій можна виконувати присвоювання контейнерів?	6	3
Тема 13. Використання класів-контейнерів бібліотеки STL: vector, list, map. ***Рекомендовані джерела:*** ***Основні:*** *1, 2, 3.* ***Додаткові:*** *5, 7.* ***Інтернет-ресурси:*** *14, 15.* **Завдання для самостійної роботи:** Вивчення матеріалів лекції та самостійне опрацювання основних літературних джерел,	4	3

1	2	3
зазначених у списку та електронних матеріалів, які надаються викладачем. Відзначити кілька особливостей, які властиві класу bitset. Доведіть на прикладах власного програмного коду обмеженість процедурно-процедурно- орієнтованої системи вводу-виводу, які успадковані від мови С. Якщо є необхідність застосувати виводу бітової множини клас cout.		

1	2	3
Тема 14. Застосування алгоритмів бібліотеки STL. ***Рекомендовані джерела:*** ***Основні:*** *1, 2, 3.* ***Додаткові:*** *4, 7.* ***Інтернет-ресурси:*** *14, 15.* **Завдання для самостійної роботи:** Вивчення матеріалів лекції та самостійне опрацювання основних літературних джерел, зазначених у списку та електронних матеріалів, які надаються викладачем. Підготувати доповідь у вигляді презентації на одну з вказаних тем: 1. В масиві обчислити різницю елементів масиву, що розташовані між першим від'ємним та другим додатним елементами. 2. Дана прямокутна матриця. Визначити кількість від'ємних елементів в тих рядках, які містять хоча б один нульовий елемент; 3. У довільній матриці обчислити кількість елементів масиву, рівних нулю; 4. В одновимірному масиві елементів, обчислити суму модулів елементів, які розташовані після першого додатного елемента. 5. У матриці обчислити суму елементів масиву, що розташовані між першим і другим додатними елементами. 6. Дана прямокутна матриця. Визначити номер рядка, в якому знаходиться найдовша серія з однакових елементів	6	3
Тема 15. Класи потоків уведення та виведення. ***Рекомендовані джерела:*** ***Основні:*** *1, 2, 3.* ***Додаткові:*** *4 - 6.* ***Інтернет-ресурси:*** *14, 15.* **Завдання для самостійної роботи:** Вивчення матеріалів лекції та самостійне опрацювання основних літературних джерел, зазначених у списку та електронних матеріалів, які надаються викладачем. Підготувати доповідь у вигляді презентації на одну з вказаних тем: 1. Застосувати прапорці формату iostream. 2. Застосувати Маніпулятори є функції (знаходяться в *iomanip.h*), які можна включати у низку	4	3

1	2	3
послідовних операцій поміщення та вилучення. !!! виключенням *setw*, зміни, внесені маніпуляторами, зберігаються до наступної модифікації.		
Тема 16. Оброблення виняткових ситуацій в C#. ***Рекомендовані джерела:*** ***Основні:*** *1, 2, 3.* ***Додаткові:*** *7, 8.* ***Інтернет-ресурси:*** *14, 15.* **Завдання для самостійної роботи:** Вивчення матеріалів лекції та самостійне опрацювання основних літературних джерел, зазначених у списку та електронних матеріалів, які надаються викладачем. Підготувати доповідь у вигляді презентації на одну з вказаних тем: 1. У якості типу перехвачуваного виключення необхідно використовувати клас Exception або його нащадків. 2. Для викидання (порушення) виключення використовується оператор throw, що має наступний синтаксис: throw виключення; 3. У найпростішому випадку для викидання загального виключення використовується наступний синтаксис: throw new Exception ("повідомлення про помилку"); 4. Якщо необхідно повторно порушити виняток, перехоплене в блоці catch, слід скористатися оператором throw без параметрів.	2	3

1	2	3
Тема 17. Розроблення DLL-бібліотек. ***Рекомендовані джерела:*** ***Основні:*** *1, 2, 3.* ***Додаткові:*** *5, 6.* ***Інтернет-ресурси:*** *14, 15* **Завдання для самостійної роботи:** Вивчення матеріалів лекції та самостійне опрацювання основних літературних джерел, зазначених у списку та електронних матеріалів, які надаються викладачем. Підготувати доповідь у вигляді презентації на одну з вказаних тем: 1. Основні характеристики БД MongoDB. 2. ODBC-інтерфейси для доступу до БД MS Access та MS SQL Server.	2	3

6. Список рекомендованих джерел

Основний

1. Настенко, Д. В. Об’єктно-орієнтоване програмування. Частина 1. Основи об’єктно-орієнтованого програмування на мові C# [Електронний ресурс] : навчальний посібник для бакалаврів напряму підготовки 6.050701 «Електротехніка та електротехнології» програми професійного спрямування «Системи управління виробництвом та розподілом електроенергії» / Д. В. Настенко, А. Б. Нестерко ; НТУУ «КПІ» – Київ : НТУУ «КПІ», 2016. – 76 с.
2. Дібрівний О.А., Гребенюк В.В. Вступ до об’єктно орієнтованого програмування C#: Навчальний посібник. – К.: Державний університет телекомунікацій, 2018, - 190с.
3. Бурлаков А. А. Об’єктно-орієнтований аналіз і проектування. Методичні рекомендації з самостійного вивчення дисципліни студентами напрямку підготовки «Програмна інженерія» / А. А. Бурлаков. – Хмельницький: ХНУ, 2017. – 136 с.

Додатковий

1. Колдовский В.В. ПО: оценка результата [Текст] / В.В. Колдовский // Компьютерное обозрение, 2006. – № 34. – С. 58 – 61.
2. Ушакова І. О. Основи системного аналізу об’єктів та процесів комп’ютеризації: навчальний посібник. Ч. 2 / І. О. Ушакова. – Х.: Вид. ХНЕУ, 2008. – 324 с.

3. Ушакова І. О. Практикум з навчальної дисципліни "Основи системного аналізу об’єктів і процесів комп’ютеризації": навчальний посібник / І. О. Ушакова, Г. О. Плеханова. – Х.: Вид. ХНЕУ, 2010. – 344 с.
4. Щедріна О.І. Алгорітмізація та програмування процедур ОБРОБКИ информации. К., 2001. - 240 с.
5. Остапов С.Е. Технології захисту інформації / Остапов С.Е., Євсєєв С.П., Король О.Г. – Чернівці: Видавничий дім «РОДОВІД», 2014. 428 с.

Інтернет ресурси

6. Інформаційно-аналітичний щотижневик “ComputerWorld”. Режим доступу: www.cw.comizdat.com
7. Журнал “Office”. Режим доступу: www.officemag.kiev.ua
8. Щотижневик “Компьютерное обозрение”. Видавничий дім ITC. Режим доступу: www.itc-ua.com
9. Щотижневик “Мой компьютер”. Видавничий дім "Мой компьютер". Режим доступу: www.mycomp.com.ua
10. Інформаційне представництво компанії Microsoft в Україні. Режим доступу: www.microsoft.com/ukraine
11. Введення в C# Режим доступу: https://programm.top/c-sharp/tutorial/introduction/
12. Приклад основних структур Даних. Режим доступу: http://radfiz.org.ua/share/oop/efi_shpore_exam/

**Курсивом зазначені джерела, що є в наявності в бібліотеці КНТЕУ*

7. Контроль та оцінювання результатів навчання:

Положення про оцінювання результатів навчання студентів і аспірантів наказ КНТЕУ №2891 від 16.09.2019р. (Електронний ресурс. Точка доступу: https://knute.edu.ua/file/NzU4MQ==/69da3a261374f213990591e6e9a812cd.pdf)

Під час вивчення дисципліни викладачем здійснюється поточний та підсумковий контроль. Поточний контроль та оцінювання передбачає:

- перевірку рівня засвоєння теоретичного матеріалу (тестування за матеріалами лекції, який здійснюється з використанням 365 Office);
- захист лабораторних робіт (проходить під час кожної лабораторної роботи);
- перевірка ходу виконання індивідуального завдання (фінальний проект);
- перевірка засвоєння матеріалу, що винесений на самостійне опрацювання під час фронтального опитування на лекції та заслуховування доповідей на обрані студентами теми.

8. Політика навчальної дисципліни:

8.1.Відвідування лекційних та лабораторних занять: відвідування лекційних та лабораторних занять є обов’язковим. Допускаються пропуски занять з таких поважних причин, як хвороба (викладачу надається копія

довідки від медичного закладу), участь в олімпіаді, творчому конкурсі тощо за попередньою домовленістю та згодою викладача за умови дозволу деканату (надаються документи чи інші матеріали, які підтверджують заявлену участь у діяльності студента).

8.2.Відпрацювання пропущених занять: відпрацювання пропущених занять є обов'язковим незалежно від причини пропущеного заняття. Лекційне заняття має бути відпрацьоване до наступної лекції на консультації викладача з використанням ПЗ 365 Office Teams. Відпрацювання лекційного матеріалу передбачає вивчення пропущеного теоретичного матеріалу та складання тесту за цим матеріалом. Лабораторне заняття відпрацьовується під час консультації викладача (розклад консультацій на сайті).

8.3.Правила поведінки під час занять: обов'язковим є дотримання техніки безпеки в комп'ютерних лабораторіях. Студенти повинні приймати активну участь в обговоренні навчально матеріалу ознайомившись з ним напередодні (навчальний матеріал надається викладачем). Мобільні пристрої дозволяється використовувати лише під час он-лайн тестування та підготовки практичних завдань в процесі заняття. Задля зручності, дозволяється використання ноутбуків та інших електронних пристроїв під час навчання в комп'ютерних аудиторіях (за взаємною згодою всіх учасників освітнього процесу)

8.4. За порушення академічної доброчесності студенти будуть притягнені до академічної відповідальності у відповідності до положення про дотримання академічної доброчесності педагогічними, науково-педагогічними, науковими працівниками та здобувачами вищої освіти КНТЕУ (Наказ КНТЕУ від 02.02.2018 №377. (Електронний ресурс. Точка доступу: https://knute.edu.ua/file/MTEyNDI=/f78c64a74cbbe5b4238729782d707efa.pdf)