

ДЕРЖАВНИЙ 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 цифрової економіки та системного аналізу

ЗАТВЕРДЖЕНО

вченою радою ДТЕУ

(пост. п. від « 30 » серпня 2022 р.)

Ректор

Анатолій МАЗАРАКІ



КРОСПЛАТФОРМЕНЕ ПРОГРАМУВАННЯ /
CROSS-PLATFORM PROGRAMMING

ПРОГРАМА /
COURSE SUMMARY

Київ 2022

**Розповсюдження і тиражування без офіційного дозволу ДТЕУ
заборонено**

Автор: С. В. Міценко, доцент кафедри цифрової економіки та системного аналізу, к.т.н., доцент

А.М.Тарасюк, асистент кафедри цифрової економіки та системного аналізу

Програму розглянуто і затверджено на засіданні кафедри цифрової економіки та системного аналізу 25.04.2022р., протокол № 3

Рецензент: В.В.Кулаженко, доцент кафедри цифрової економіки та системного аналізу, к.е.н., доцент

М. Г. Шарафутдінов, директор з розвитку компанії «CenterResearch&Development», бізнес-аналітик

**КРОС-ПЛАТФОРМНЕ ПРОГРАМУВАННЯ /
CROSS-PLATFORM PROGRAMMING**

**ПРОГРАМА /
COURSE SUMMARY**

ВСТУП

Програма дисципліни «Крос-платформне програмування» призначена для студентів першого (бакалаврського) рівня вищої освіти ДТЕУ денної форми навчання галузі знань 12 «Інформаційні технології», спеціальності 124 «Системний аналіз» освітньої програми «Інформаційні технології та бізнес-аналітика (Data Science)»

Програму підготовлено відповідно до Стандартів вищої освіти України зі спеціальності 124 «Системний аналіз» та освітньо-професійної програми ДТЕУ «Інформаційні технології та бізнес-аналітика (Data Science)» першого (бакалаврського) рівня вищої освіти.

Програма складається з таких розділів:

1. Мета, завдання та предмет дисципліни.
2. Передумови вивчення дисципліни як вибіркової компоненти освітньої програми.
3. Результати вивчення дисципліни.
4. Зміст дисципліни.
5. Список рекомендованих джерел.

1. МЕТА, ЗАВДАННЯ ТА ПРЕДМЕТДИСЦИПЛІНИ

Метою вивчення дисципліни «Крос-платформне програмування» є надання фундаментальних теоретичних знань і набуття практичних навичок з питань аналізу, підготовки та розробки програмного забезпечення що коректно працює на різних платформах.

Завданням вивчення дисципліни «Крос-платформне програмування» є надання студентам ґрунтовних знань в області розробки програмного забезпечення, вивчення алгоритмів розробки забезпечення, роботи із структурами даних, реалізації патернів програмування роботою з системами контролю версій.

Предметом вивчення дисципліни є основні підходи до крос-платформної розробки програмного забезпечення..

2. ПЕРЕДУМОВИ ВИВЧЕННЯ ДИСЦИПЛІНИ ЯК ВИБІРКОВОЇ КОМПОНЕНТИ ОСВІТНЬОЇ ПРОГРАМИ

знання

- основ інформаційних технологій (операційна система *Windows*, бази даних, доступ до веб-ресурсів);
- основ програмування (концепції програмування, парадигми, алгоритми, структури даних);

вміння

- вільно працювати з офісними додатками *Microsoft Word, Microsoft Excel, Microsoft PowerPoint*.

3. РЕЗУЛЬТАТИ ВИВЧЕННЯ ДИСЦИПЛІНИ

Дисципліна «Крос-платформене програмування», як обов'язкова компонента освітньої програми «Інформаційні технології та бізнес-аналітика (Data Science)» забезпечує оволодіння студентами загальними та фаховими компетентностями і досягнення ними програмних результатів навчання за відповідними освітньо-професійними програмами:

«Інформаційні технології та бізнес-аналітика (Data Science)» (ОС бакалавр)

Номер в освітній програмі	Зміст компетентності	Номер теми, що розкриває зміст компетентності
<i>Загальні компетентності за освітньою програмою</i>		
K01.	Здатність до абстрактного мислення, аналізу та синтезу	1-8,10-12
K02.	Здатність застосовувати знання у практичних ситуаціях	3-10
K07.	Здатність до пошуку, оброблення та аналізу інформації з різних джерел	4-8,12-14
<i>Спеціальні (фахові, предметні) компетентності за освітньою програмою</i>		
K17.	Здатність використовувати системний аналіз як сучасну міждисциплінарну методологію, що базується на прикладних математичних методах та сучасних інформаційних технологіях і орієнтована на вирішення задач аналізу і синтезу технічних, економічних, соціальних, екологічних та інших складних систем.	3-7
K21.	Здатність формулювати задачі оптимізації при проектуванні систем управління та прийняття рішень, а саме: математичні моделі, критерії оптимальності, обмеження, цілі управління; обирати раціональні методи та алгоритми розв'язання задач оптимізації та оптимального керування.	4-9
K22.	Здатність до комп'ютерної реалізації математичних моделей реальних систем і процесів; проектувати, застосовувати і супроводжувати програмні засоби моделювання, прийняття рішень, оптимізації, обробки інформації, інтелектуального аналізу даних.	5-10
K23.	Здатність використовувати сучасні інформаційні технології для комп'ютерної реалізації математичних моделей та прогнозування поведінки конкретних систем а саме: об'єктно-орієнтований підхід при проектуванні складних систем різної природи, прикладні математичні пакети, застосування баз даних і знань.	8-14
K24.	Здатність організувати роботу з аналізу та проектування складних систем, створення відповідних інформаційних технологій та програмного забезпечення.	3-8

<i>Програмні результати навчання за освітньою програмою</i>		
ПР07.	Знати основи теорії оптимізації, оптимального керування, теорії прийняття рішень, вміти застосовувати їх на практиці для розв'язування прикладних задач управління і проектування складних систем.	4-10
ПР08.	Володіти сучасними методами розробки програм і програмних комплексів та прийняття оптимальних рішень щодо складу програмного забезпечення, алгоритмів процедур і операцій.	3-8
ПР09.	Вміти створювати ефективні алгоритми для обчислювальних задач системного аналізу та систем підтримки прийняття рішень.	3-6
ПР10.	Знати архітектуру сучасних обчислювальних систем і комп'ютерних мереж.	1-6
ПР11.	Знати і вміти застосовувати на практиці системи управління базами даних і знань та інформаційні системи.	3-9
ПР13.	Проектувати, реалізовувати, тестувати, впроваджувати, супроводжувати, експлуатувати програмні засоби роботи з даними і знаннями в комп'ютерних системах і мережах.	8-14
ПР14.	Розуміти і застосовувати на практиці методи статистичного моделювання і прогнозування, оцінювати вихідні дані.	4-7

4. ЗМІСТ ДИСЦИПЛІНИ

Тема 1. Принципи крос-платформного підходу

Підходи до розробки програмних продуктів. Крос-платформне програмування. Порівняння нативної та крос-платформної розробки. Переваги та недоліки обох підходів.

Розробка програмного забезпечення на мові програмування C# на базі платформ .Net Core та платформи мобільної розробки Xamarin. Переваги застосування мови C# для крос-платформної розробки.

Розробка програмного забезпечення на основі мови програмування Python. Фреймворк Kivy для ефективною розробки крос-платформного програмного забезпечення.

Список рекомендованих джерел

Основний: 1[с.12-32],2[с.8-20],4[с.21-32].

Додатковий: 6[с.20-44],8[с.28-40],9[с.14-32].

Інтернет-ресурси: 4,5

Тема 2. Підходи до розробки крос-платформного програмного забезпечення

Паридигми розробки програмного забезпечення. Об'єктно-орієнтована парадигма розробки як найзручніший підхід до розробки додатків. Основні принципи об'єктно-орієнтованої парадигми.

Поняття інкапсуляції. Підходи до визначення інкапсуляції. Інкапсуляція та приховування. Модифікатори доступу як елемент приховування даних.

Поняття спадкування. Критика спадкування. Переваги застосування спадкування у системах. Підходи до розробки ієрархії. Поняття Upcasting та Downcasting.

Поняття поліморфізму. Поліморфізм та статична типізація. Види поліморфізму. Поліморфізм типів. Поліморфізм методів. AdHoc поліморфізм.

Список рекомендованих джерел

Основний: 1[с.22-56],2[с.28-40],4[с.22-40].

Додатковий: 5[с.22-56],6[с.18-32],10[с.24-32].

Інтернет-ресурси: 4,5

Тема 3. Типи даних у мові C# Застосування алгоритмів у крос-платформному програмному забезпеченні. Поняття Big O

Типи даних у мові C#. Базові та посилальні типи даних, їх основні відмінності. Базові типи для роботи з числами. Логічні значення. Текстові типи даних. Поняття явного та неявного приведення типів. Логічні операції

Поняття умовних конструкцій. Види умовних конструкцій. If...else. Тернарний оператор та вимоги до його застосування.

Поняття циклу. Види циклів. Цикли з передумовою. Цикли з після умовою.

Поняття алгоритму. Види алгоритмів. Класифікація алгоритмів. Складність алгоритмів та концепція Big O. Підходи до спрощення алгоритмів.

Список рекомендованих джерел

Основний: 1[с.32-56],2[с.36-49],4[с.28-48].

Додатковий: 5[с.27-46],6[с.22-38],10[с.38-45].

Інтернет-ресурси: 4,5

Тема 4. Типи даних DateTime. Робота з датою та часом у C#. Типи даних String. Робота з текстовими даними

Тип даних DateTime. Застосування дати та часу у крос-платформному програмному забезпеченні. Відлік часу. Види часу.

Форматування дати та часу. UI інтерфейси для вводу та виводу дати і часу. Календар. Сортування дати і часу. Особливі випадки сортування дати та часу

Тип даних String. Операції з текстовими даними у крос-платформному програмному забезпеченні. Операція з рядками. Форматування та інтерполяція строк. Клас StringBuilder

Форматування тексту. Регулярні вирази. Правила застосування регулярних виразів та вимоги до них.

Список рекомендованих джерел

Основний: 1[с.32-56],2[с.42-69],3[с.48-78].

Додатковий: 5[с.21-27],6[с.39-54],10[с.46-62].

*Інтернет-ресурси:*4,5

Тема 5. Класи та інтерфейси у крос-платформному програмному забезпеченні. Призначення та специфіка

Поняття класу у програмування. Види класів. Абстрактні класи та їх можливості. Конструктори класів .Приватний та статичний конструктор. Заборона спадкування класу. Види спадкувань

Поняття інтерфейсу у програмування. Можливості застосування інтерфейсів. Інтерфейси і абстрактні класи. Реалізація інтерфейсів.

Поняття Узагальнень (Generics), та їх застосування. Узагальнення у методах. Обмеження узагальнень.

Список рекомендованих джерел

Основний: 1[с.38-60],2[с.56-78],4[с.58-78].

Додатковий: 5[с.21-42],6[с.45-76],8[с.32-56].

*Інтернет-ресурси:*4,5

Тема 6. Структура крос-платформного проекту

Найпоширеніші патерни розробки додатків. Підходи до поділу додатку на рівні. Принципи SOLID,KISS,DRY.

Патерн MVC та його особливості. Роль моделі у додатку. Робота з контролерами. Проектування візуального дизайну

Патерн MVVM та його особливості. Розробка моделей для системи. Проектування візуального представлення. Розробка рівня View-Model.

Мова UML. UML діаграми. Види UML діаграм. Діаграма класів. Діаграма активностей.

Список рекомендованих джерел

Основний: 1[с.32-56],2[с.42-69],3[с.48-78].

Додатковий: 5[с.21-27],6[с.39-54],10[с.46-62].

*Інтернет-ресурси:*4,5

Тема 7. Робота з даними мовою С#. Структури даних мови С#. Робота з базами даних.

Поняття даних у програмному забезпеченні. Види обробки даних. Вимоги до зберігання даних.

Поняття структури даних. Основні структури даних. Масиви. Їх види, застосування. Масив як базова структура даних. Список як удосконалений вид масиву. Зв'язаний список та його напрями застосування.

Граф як спосіб класифікації даних. Дерево як елемент для створення алгоритмів та ієрархій. Стек та Черга як спосіб контролю входу та виходу даних. Хеш-таблиця та її застосування.

Поняття бази даних. Види баз даних. NoSQL бази даних та їх класифікація. База даних SQL. Мова запитів SQL та T-SQL.

Поняття децентралізованих сховищ даних. Блокчейн як новий підхід до зберігання та забезпечення даних. Поняття DTO (Data Transfer Object).

Вимоги до підключення сторонніх сервісів та баз даних. Поняття ORM

Список рекомендованих джерел

Основний: 1[с.32-56],2[с.36-49],4[с.28-48].

Додатковий: 5[с.27-46],6[с.22-38],10[с.38-45].

Інтернет-ресурси: 1,2

Тема 8. Робота з базами даних у C#. Специфіка роботи з NoSQL базами даних

Поняття бази даних. Види баз даних. NoSQL бази даних та їх класифікація. База даних SQL. Мова запитів SQL та T-SQL.

Поняття децентралізованих сховищ даних. Блокчейн як новий підхід до зберігання та забезпечення даних. Поняття DTO (Data Transfer Object).

Вимоги до підключення сторонніх сервісів та баз даних. Поняття ORM

Список рекомендованих джерел

Основний: 1[с.32-56],2[с.36-49],4[с.28-48].

Додатковий: 5[с.27-46],6[с.22-38],10[с.38-45].

Інтернет-ресурси: 1,2

Тема 9. Додаткові можливості об'єктно орієнтованого програмування

Поняття делегатів. Роль делегатів у реалізації подій системи. Види подій. Обробка подій у об'єктно-орієнтованому програмуванні.

Оператори. Оператори у структурах. Модифікатори у операторах. Перегрузка операторів. Перетворення типів.

Виключення. Види та підходи до обробки. Конструкція try...catch...finally. Створення виключень у системі шляхом спадкування. Створення виключень з допомогою конструкції throw new exception.

Індексатори. Методи розширення. Підходи до застосування методів розширення. Методи розширення

Часткові класи та методи. Підходи до застосування часткових класів. Анонімні методи. Робота з кортежами.

Список рекомендованих джерел

Основний: 1[с.32-56],2[с.42-69],3[с.48-78].

Додатковий: 5[с.21-27],6[с.39-54],10[с.46-62].

Інтернет-ресурси: 4,5

Тема 10. LINQ та його застосування. F# та його можливості у C#

LINQ. Ключові слова. Директиви. Підходи до написання запитів. Проекція даних. Фільтрація даних. Сортування. Об'єднання перерізів. Агрегатні операції та групування. Делегати і запити у LINQ.

Рефлексія. Асинхронне програмування. Pattern matching. Патерн типів. Паттерн властивостей. Паттерни кортежів. Позиційний патерн. Реляційний та логічний патерни. Паттерни списків.

F# як доповнення до мови C#. Підходи до застосування функціональної парадигми у об'єктно-орієнтованому програмуванні. Типи даних у F#. Основні функції у F#. F# і ООП.

Список рекомендованих джерел

Основний: 1[с.32-56], 2[с.42-69], 3[с.48-78].

Додатковий: 5[с.21-27], 6[с.39-54], 10[с.46-62].

Інтернет-ресурси: 4, 5

Тема 11. Коректність і безпека крос-платформного програмного забезпечення

Поняття коректності у крос-платформному програмному забезпеченні. Мінімізація впливу користувача.

Некоректна поведінка програмного забезпечення. Поняття виключення у програмних засобах. Обробка виключень. Фільтр виключень. Користувацькі виключення.

Поняття безпеки програмного забезпечення. Ролі у системі. Аутентифікація та авторизація у системі. Вимоги до паролів та шифрування.

Список рекомендованих джерел

Основний: 1[с.12-25], 2[с.67-82], 4[с.49-52].

Додатковий: 5[с.38-45], 6[с.40-56], 10[с.46-65].

Інтернет-ресурси: 4, 5

Тема 12. Патерни у крос-платформному програмному забезпеченні

Поняття патерну програмування. Їх переваги та недоліки. Класифікації патернів програмування.

Породжувальні патерни програмування. Фабричний метод. Абстрактна фабрика. Будівельник. Прототип. Одинак.

Поведінкові патерни програмування. Стратегія. Ланцюжок обов'язків. Посередник. Команда. Ітератор. Спостерігач. Стан. Відвідувач. Знімок. Шаблонний метод.

Структурні патерни програмування. Адаптер. Міст. Компонувальник. Декоратор. Фасад. Замісник. Легковаговик.

Список рекомендованих джерел

Основний: 1[с.12-25], 2[с.67-82], 4[с.49-52].

Додатковий: 5[с.38-45], 6[с.40-56], 10[с.46-65].

Тема 13. Логіювання та документування коду. Code convention.

Поняття логіювання коду. Створення логів у системі. Правила створення логів. Застосування логів для мінімізації шкоди від зовнішніх загроз.

Поняття Code convention. Підходи до іменування у програмному забезпеченні. Вимоги до іменування класів та інтерфейсів. Різниця у іменуванні приватних та публічних полів. Іменування структур.

Види іменувань. Угорське іменування та його недоліки.

Поняття документування коду. Підходи до документування коду. Хороші та погані практики. Документування бізнес-логіки. Документування класів. Правила документування методів та виключень.

Список рекомендованих джерел

Основний: 1[с.75-82],2[с.85-97],4[с.72-90].

Додатковий: 5[с.48-60],6[с.58-90],10[с.75-89].

*Інтернет-ресурси:*4,5

Тема 14. Тестування крос-платформного програмного забезпечення

Поняття тестування коду. Поняття тесту. Правила тестування. Методи тестування. Види тестів. Оптимізації тестів та їх навантаження. Поняття покриття системи тестами.

Тестування за знанням системи. Тестування чорної скриньки (blackbox). Тестування білої скриньки (whitebox). Тестування сірої скриньки (greybox)

Тестування за ступенем ізольованості компонентів. Компонентне (модульне) тестування (component/unittesting). Інтеграційне тестування (integrationtesting). Системне тестування (system/end-to-endtesting).

Тестування за часом проведення. Альфа-тестування (alphatesting). Тестування при прийманні або Димове тестування (smoketesting). Тестування нової функціональності (newfeaturetesting). Регресивне тестування (regressiontesting). Тестування при здачі (acceptancetesting). Бета-тестування (betatesting)

Тестування за об'єктом. Функціональне тестування (functionaltesting) Тестування зручності використання або Юзабіліті-тестування (usabilitytesting). Тестування інтерфейсу користувача (UI testing). Тестування безпеки (securitytesting). Тестування локалізації (localizationtesting). Тестування сумісності (compatibilitytesting).

Тестування продуктивності (performancetesting). Навантажувальне тестування (loadtesting). Стрес-тестування (stresstesting). Тестування стабільності (stability/endurance/soaktesting).

Список рекомендованих джерел

Основний: 1[с.75-82],2[с.85-97],4[с.72-90].

Додатковий: 5[с.48-60],6[с.58-90],10[с.75-89].

*Інтернет-ресурси:*4,5

5. СПИСОК РЕКОМЕНДОВАНИХ ДЖЕРЕЛ

Основний

1. PRICE, Mark J. *C# 8.0 and .NET Core 3.0—Modern Cross-Platform Development: Build applications with C#, .NET Core, Entity Framework Core, ASP.NET Core, and ML.NET using Visual Studio Code*. Packt Publishing Ltd, 2019.
2. ALBAHARI, Joseph. *C# 9.0 in a Nutshell*. " O'ReillyMedia, Inc.", 2021.
3. PRICE, Mark J. *C# 9 and .NET 5—Modern Cross-Platform Development: Build intelligent apps, websites, and services with Blazor, ASP.NET Core, and Entity Framework Core using Visual Studio Code*. Packt Publishing Ltd, 2020.
4. CLEARY, Stephen. *Concurrency in C# Cookbook: Asynchronous, Parallel, and Multithreaded Programming*. O'Reilly Media, 2019.
5. DOOLEY, John F.; DOOLEY. *Software Development, Design and Coding*. Apress, 2017.

Додатковий

6. BILGIN, Can. *Mastering Cross-Platform Development with Xamarin*. PacktPublishingLtd, 2016
7. TURNER, Ryan. *C#: The Ultimate Beginner's Guide to Learn C# Programming Step by Step*. Publishing Factory, 2020.
8. SOLIS, Daniel; SCHROTENBOER, Cal. *Illustrated C# 7: The C# Language Presented Clearly, Concisely, and Visually*. Apress, 2018.
9. SKEET, Jonathan. *C# in Depth*. Simonand Schuster, 2019.
10. ALBAHARI, Joseph; ALBAHARI, Ben. *C# 10 Pocket Reference*. " O'Reilly Media, Inc.", 2022.
11. JAMRO, Marcin. *C# Data Structures and Algorithms: Explore the possibilities of C# for developing a variety of efficient applications*. Packt Publishing Ltd, 2018.
12. RUMPE, Bernhard. *Agile modeling with UML: code generation, Testing, Refactoring*. Cham: Springer, 2017.
13. XU, Jack. *Practical C# Charts and Graphics: Advanced Chart and Graphics Programming for Real-World .NET Applications*. UniCAD, 2019.

Інтернет-ресурси

1. Типи даних. URL: https://uk.wikipedia.org/wiki/%D0%A2%D0%B8%D0%BF_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85
2. Структура даних. URL: https://uk.wikipedia.org/wiki/%D0%A1%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%B0_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85
3. Патерни програмування. URL: <https://refactoring.guru/uk/design-patterns>
4. Документація на C#. URL: <https://docs.microsoft.com/ru-ru/dotnet/csharp/>

* Курсивом виділені джерела, що є у бібліотеці ДТЕУ, або наявні повнотекстові електронні версії джерел.