

**Вищий навчальний заклад  
Університет економіки та права "КРОК"  
Коледж економіки, права та інформаційних технологій**

**ЖОВТЯК І.В., ДОБРИШИН Ю.Є.**

**МЕТОДИЧНІ ВКАЗІВКИ ЩОДО ВИКОНАННЯ ЗАВДАНЬ  
ТЕХНОЛОГІЧНОЇ ПРАКТИКИ**

**Частина 2**

Для студентів спеціальностей 121 «Інженерія програмного забезпечення»  
та 122 «Комп'ютерні науки»

**Київ 2019 р.**

## **УДК 004.658.2**

Розглянуто на засіданні циклової комісії з інформаційних технологій протокол № 1 від «28» серпня 2019 р Рекомендовано до видання методичною радою Коледжу економіки, права та інформаційних технологій «Університет економіки та права «КРОК» протокол №1 від «30» серпня 2019 р.

### **Автори:**

1. І.В. Жовтяк, викладач Коледжу економіки, права та інформаційних технологій «Університет економіки та права «КРОК».

2. Ю.Є. Добришин, кандидат технічних наук, доцент кафедри комп'ютерних наук Навчально-наукового інституту інформаційних та комунікаційних технологій «Університет економіки та права «КРОК».

[Текст]: Методичні вказівки щодо виконання завдань технологічної практики [ І.В. Жовтяк, Ю.Є. Добришин]; Університет економіки та права «КРОК» – Київ - 2019. – 14с.

Методичні рекомендації щодо виконання завдань технологічної практики містять постановку задачі, вимоги до проекту, завдання та вказівки щодо їх виконання, перелік тем для проектів, критерії підсумкового оцінювання, список літератури та інформаційних ресурсів.

Видання призначене для студентів спеціальностей 121 «Інженерія програмного забезпечення» та 122 «Комп'ютерні науки»

## **РОЗГЛЯНУТО І СХВАЛЕНО**

Педагогічною радою Коледжу економіки, права та інформаційних технологій  
Протокол № 1 від «30» серпня 2019 р.

УДК 004.658.2

©Жовтяк І.В. 2019р.

©Добришин Ю.Є. 2019р.

©Коледж економіки, права та інформаційних технологій, 2019 р.

©Університет економіки та права «КРОК» 2019 р

## ЗМІСТ

1. ВСТУП .....	4
2. Постановка задачі.....	6
3. Вимоги до проекту .....	6
3.1. Робота з даними.....	6
3.2. Дизайн застосування.....	7
3.3. Організація роботи.....	7
4. Завдання та вказівки щодо виконання .....	8
4.1. Завдання 1 (1 тиждень).....	8
4.2. Завдання 2 (2-4 тиждень).....	8
5. Перелік тем .....	8
6. КРИТЕРІЇ ОЦІНЮВАННЯ ВИКОНАННЯ технологічної практики.....	11
6.1. Загальні положення.....	11
6.2. Перелік вимог до проекту технологічної практики.....	12
6.2.1. Об'єктно-орієнтоване проектування та робота з даними .....	12
6.2.2. Дизайн застосування.....	13
7. Перелік навчальної літератури та посилань на ресурси.....	13

## 1. ВСТУП

Основною метою технологічної практики є закріплення студентами знань, які були раніше отримані під час вивчення дисциплін “Об’єктно-орієнтоване програмування” і “Web-технології та Web-дизайн”, придбання практичних навичок створення власних програмних застосунків.

Для досягнення цієї мети розглядаються наступні завдання:

формування у студентів практичних умінь та навичок розробки програмного забезпечення у різних предметних галузях, ефективного використання призначених для цього засобів;

закріплення на практиці отриманих знань з теоретичних основ об’єктно-орієнтованого програмування, web-технологій та , розвиток практичних умінь створення desktop та web-застосунків тощо;

ознайомлення із існуючими технологіями, методами та засобами розробки програмного забезпечення, удосконалення навичок використання інтегрованих середовищ програмування та організації процесу розробки, вивчення та практичне використання методик роботи у групі програмістів;

формування умінь самостійно працювати над поставленою задачею, використовувати для об’єктно-орієнтованого проектування необхідні структури даних, шаблони, патерни, здійснювати розробку та пошук алгоритмів обробки даних;

формування умінь використання та розробки засобів для обробки текстових документів, електронних таблиць, презентацій, графічних зображень.

У результаті проходження навчальної практики студент навчиться:

здійснювати об’єктно-орієнтоване проектування та розробляти найпростіші desktop- та web-застосунки у різних предметних галузях, ефективно використовувати призначені для цього засоби;

знаходити методи розв’язку типових задач з об’єктно-орієнтованого програмування та web-технологій, інших дисциплін навчального плану;

вміти використовувати патерни, списки, об’єктно-орієнтоване відображення, системи керування базами даних, використовувати файли, графіку та інше;

використовувати типові шаблони побудови програм з графічним інтерфейсом, засоби інтегрованих середовищ розробки.

використовувати засоби організації процесу розробки програмного забезпечення та командної роботи;

використання текстових редакторів, електронних таблиць, редакторів комп'ютерних презентацій для створення документів;

демонструвати техніко-експлуатаційні характеристики, конструктивні особливості і призначення розробленого програмного застосування з використанням звіту про результати практики.

Як метод навчальної діяльності використано метод проектів. На протязі всією практики студенти знайомляться із зразком виконання проекту, а потім застосовують отримані знання та навички для розробки проекту за індивідуальним завданням.

## **2. ПОСТАНОВКА ЗАДАЧІ**

Проект з технологічної практики з дисципліни «Об'єктно-орієнтоване програмування» реалізується з використанням однієї з технологій Windows Forms, WPF, ASP.NET Web Forms, ASP.NET Framework MVC, ASP.NET Core MVC в середовищі Visual Studio.

Студент повинен розробити застосування відповідно до своєї теми. У проекті має зберігатись структурована інформація в базі даних і відображатись на сторінках сайту - каталог товарів (або послуг) і т.д.

Для спрощення процесу інформаційного наповнення сторінок і пошуком графічного матеріалу, можна використати, як аналог, готовий шаблон, або будь-який існуючий сайту зі змінами, які відповідають завданням проекту.

## **3. ВИМОГИ ДО ПРОЕКТУ**

### **3.1. Робота з даними**

1. Застосування повинно бути реалізовано за тривірневою клієнт-серверною архітектурою, містити адмін- та клієнтську частини.

2. Для ведення даних використовувати засоби SQL Server.

3. В застосуванні потрібно використати не менше трьох таблиць, одна з яких має бути асоційованою.

4. Для роботи з даними необхідно використовувати об'єктно-реляційне відображення за технологією Entity Framework (ORM) на основі Code First або Data First. Альтернативно, може бути використана технологія NHibernate.

5. Для відображення схеми БД використовувати POCO-класи моделі даних, даних – контекст даних.

6. Для реалізації CRUD – операцій, вибірки та інших запитів використовувати LINQ або л-вирази.

7. Засоби адміністрування користувачів повинні бути побудовані на основі ASPNET Membership чи ASPNET Identity на основі використання ролей.

8. В клієнтській частині повинні використовуватись засоби валідації введених користувачем даних;

9. Система повинна містити конфігураційні файли, які задають параметри програми (наприклад, з даними про кольори тексту, шрифти, розміри вікон, малюнків, тощо). Конфігураційний файл розміщується на сервері, і може бінарним, XML-файлом або стандартним файлом .config. Рядок з'єднання – у конфігураційному файлі.

10. У Windows WPF- чи Web-Forms-застосуваннях потрібно використовувати Web-сервіси (додатково).

11. У Web - застосуваннях потрібно використовувати маршрутизацію.

### **3.2. Дизайн застосування**

1. Верхній блок – слайдер зображень (або відео) основне і меню сайту. Пункти меню розташовані горизонтально з відступом.

2. Обов'язковими є сторінки з категоріями об'єктів, детальною характеристикою об'єкту, оберненим зв'язком і авторизацією користувачів.

3. На сторінці з даними повинна бути можливість впорядкування, фільтрації та пошуку.

4. Сторінки веб-застосування повинні займати всю ширину вікна і коректно відображатися у всіх браузерах.

5. Дизайн сторінок має бути адаптивний. При зміні розмірів вікна браузера блок контенту повинен розтягуватися на весь доступний йому простір, за винятком лівого меню (якщо є), шапки і breadcrumbs.

6. На сторінці «Контакти» має бути карта із зазначенням місця розташування об'єкту, GPS координати і дані повинні братися з бази.

7. При верстці не можна використовувати фрейми. Таблиці можна використовувати лише тоді, коли дані є табличними за своєю структурою.

### **3.3. Організація роботи**

1. Завдання та хід його виконання реєструється в системі управління проектом, для кожного завдання визначається термін його виконання.

2. Необхідною умовою зарахування роботи є використання системи контролю версій. Перевірятись будуть тільки ті проекти, які добавлені на TFS на початку практики і до яких надано доступ з правами адміністратора вказаному

## 4. ЗАВДАННЯ ТА ВКАЗІВКИ ЩОДО ВИКОНАННЯ

### 4.1. Завдання 1 (1 тиждень)

Виконати навчальний проект за покроковою інструкцією “Початок роботи ASP.NET Framework MVC”:

<https://docs.microsoft.com/ru-ru/aspnet/mvc/overview/getting-started/introduction/getting-started>

### 4.2. Завдання 2 (2-4 тиждень)

Для роботи з даними використати об’єктно-реляційний мапінг за технологією Entity Framework. Відповідно до завдань свого проекту використати підхід Code First або Data First.

Інструкції щодо використання технології Entity Framework:

*підхід Code First*

<https://docs.microsoft.com/ru-ru/aspnet/mvc/overview/getting-started/getting-started-with-ef-using-mvc/>

*підхід Data First*

<https://docs.microsoft.com/ru-ru/aspnet/mvc/overview/getting-started/database-first-development/>

1. Авторизація та аутентифікація в застосуваннях виконати за інструкцією

<https://docs.microsoft.com/ru-ru/aspnet/mvc/overview/security/>

2. Інструкція з ASP.NET Core 2.1

<https://docs.microsoft.com/en-us/aspnet/core/tutorials/first-mvc-app/?view=aspnetcore-2.1>

3. Інструкція з ASP.NET Core 3.1

<https://docs.microsoft.com/en-us/aspnet/core/tutorials/first-mvc-app/start-mvc?view=aspnetcore-3.1&tabs=visual-studio>

<https://docs.microsoft.com/en-us/aspnet/core/getting-started/?view=aspnetcore-3.1&tabs=windows>

<https://docs.microsoft.com/en-us/aspnet/core/tutorials/first-mvc-app/?view=aspnetcore-3.1>

## 5. ПЕРЕЛІК ТЕМ

1. Розробка веб-сайту "Бронювання місць в готелі" засобами ASP.NET.



2. Розробка веб-сайту "Замовлення столиків в ресторані" засобами ASP.NET.
3. Розробка веб-сайту "Замовлення піци" засобами ASP.NET.
4. Розробка веб-сайту "Замовлення послуг СТО" засобами ASP.NET.
5. Використання засобів Code First Entity Framework та ASP.NET MVC для розробки веб-застосувань (на прикладі сайту "Подарункові набори").
6. Використання засобів Data First Entity Framework та ASP.NET MVC для розробки веб-застосувань.
7. Розробка навчального інформаційного каналу для студентів університету.
8. Використання WPF та веб-сервісів для front end розробки (на прикладі застосування "Мережа "Смачна кава. Каса").
9. Використання WPF та веб-сервісів для front end розробки (на прикладі застосування "Мережа "Смачна кава. Адміністратор").
10. Використання ORM на основі Entity Framework для back end розробки (на прикладі застосування "Мережа "Смачна кава").
11. Використання технологій Windows Forms та веб-сервісів для Front end розробки (на прикладі застосування "Торгівельна мережа. Каса").
12. Використання Windows Forms та веб-сервісів для front end розробки (на прикладі застосування "Торгівельна мережа. Склад").
13. Використання ORM для розробка back end частини клієнт-серверного застосування (на прикладі застосування "Торгівельна мережа").
14. Використання технологій WPF та веб-сервісів для front end розробки (на прикладі застосування "Туристична агенція. Менежер з продажу").
15. Використання Windows Forms та веб-сервісів для front end розробки (на прикладі застосування "Туристична агенція. Адміністратор").
16. Використання ORM для розробки Back end частини клієнт-серверного застосування (на прикладі застосування "Туристична агенція.>").
17. Розробка застосування "Бронювання місць в готелі" засобами ASP.NET.
18. Розробка веб-сайту "Замовлення столиків в ресторані" засобами ASP.NET.
19. Розробка веб-сайту "Бронювання авіквітоків" засобами ASP.NET.
20. Розробка кросплатформених застосувань за технологією ASP.NET Core ( на прикладі розробки веб-сайту "Блог").
21. Розробка веб-сайту "MyInstagram" засобами ASP.NET.
22. Використання Web-сервісів в WPF-застосуваннях на прикладі застосування "Аукціон творів мистецтва".
23. Порівняння засобів технологій WPF та Windows Forms для відображення графічного та мультимедійного контенту на прикладі розробки Windows застосування навчальної системи.
24. Порівняння засобів технологій WPF та Windows Forms для відображення

графічного та мультимедійного контенту на прикладі розробки WPF застосування навчальної системи.

25. Використання WPF для розробки системи тестування знань з графічним та мультимедійним контентом.

26. Використання ORM та Entity Framework на прикладі застосування "Система тестування знань".

27. Синхронізація потоків в розподілених Desktop застосуваннях на прикладі застосування "Театральна каса".

28. Дослідження можливості використання патерну Singleton в розподілених Desktop застосуваннях.

29. Реалізації CRUD операцій засобами LINQ в трирівневому застосуванні "Система відстеження завдань".

30. Використання механізмів пізнього завантаження та відкладеного виконання запитів в розподілених застосуваннях.

31. Використання лямбда-виразів в представленнях front end частини ASP.NET MVC застосування "Система відстеження завдань".

32. Робота з мультимедійним контентом у WPF застосуванні "Бібліотека контенту" (текст, аудіо, відео)

33. Реалізації універсальних CRUD операцій за допомогою збережених процедур та механізму Reflection.

34. Використання веб-сервісів у back end частині застосування "Планування подорожей".

35. Використання веб-сервісів у front end частині Web-застосування "Планування подорожей".

36. Використання RESTFull веб-сервісів в ASP.NET MVC (на прикладі розробки back end частини Web-застосування "Інтернет-магазин").

37. Використання RESTFull веб-сервісів в ASP.NET MVC (на прикладі розробки front end частини Web-застосування "Інтернет-магазин").

38. Використання веб-сервісів ASP.NET (на прикладі front end частини Web-застосування "Інтернет-аукціон").

39. Використання веб-сервісів ASP.NET (на прикладі back end частини Web-застосування "Інтернет-аукціон").

40. Front end частина WPF застосування автоматизації роботи автомобільного салону. Back end частина застосування автоматизації роботи автомобільного салону засобами MVC.

41. Автоматизація служби доставки інтернет-магазину мобільних телефонів.

42. Розробка клієнт-серверного застосування "Бронювання місць в кінотеатрі" засобами ASP.NET.

43. Розробка клієнт-серверного застосування "Бронювання місць в потязі "

засобами ASP.NET.

44. Розробка Back end частини клієнт-серверного застосування замовлення та доставки замовлення з японського ресторану.

45. Розробка front end частини клієнт-серверного застосування замовлення та доставки замовлення з японського ресторану.

46. Розробка клієнт-серверного застосування “Художня галерея” засобами ASP.NET.

47. Розробка клієнт-серверного застосування “Доставка десертів” засобами ASP.NET.

48. Використання JSON + jquery для реалізації асинхронних веб-застосовувань

49. Розробка застосування "Служба виклику таксі " засобами ASP.NET.

50. Розробка online застосування "Замовлення канцелярських товарів”.

51. Використання JSON + jquery для реалізації асинхронних веб-застосовувань.

52. Система “Абітурієнт”.

53. Використання JSON для реалізації API інтерфейсу Web-застосовувань

54. Використання JSON + jquery для реалізації асинхронних веб-застосовувань

## **6. КРИТЕРІЇ ОЦІНЮВАННЯ ВИКОНАННЯ ТЕХНОЛОГІЧНОЇ ПРАКТИКИ**

### **6.1. Загальні положення**

Основою мета перевірки виконання завдань технологічної практики – виявлення здатності студента застосовувати одержані теоретичні знання на практиці.

Оцінювання здійснюється за модульно-рейтинговою системою. Максимальний рейтинг кожного студента складається з оцінок балів завдань задач, які передбачені практикою.

Оцінка “відмінно” ставиться, якщо результати задовільняють перелік вимог до проекту і вони виконані в повному обсязі, студент демонструє знання використаних методів на рівні 90–100%.

Оцінка “добре” - результати задовільняють перелік вимог до проекту і вони виконані в повному обсязі, але студент демонструє знання використаних методів на рівні 75–90 %.

Оцінка “задовільно” – результати задовільняють частково задовільняють перелік вимог до проекту і вони виконані не в повному обсязі,, студент демонструє знання використаних методів на рівні 50–75 %.

Оцінка “незадовільно” – якщо студент не виконав завдання щодо створення проекту, не відповідає на теоретичні питання, які відносяться до теми практики.

Рейтинг виставляється під час захисту і демонстрації результатів виконання завдань з практики, і переводиться в оцінку за національною шкалою та за шкалою ECTS відповідно до таких критеріїв:

90–100 балів — «відмінно» / A

80–89 бали — «добре» / B

70–79 бали — «добре» / C

60–69 бали — «задовільно» / D

50–59 балів — «задовільно» / E

Менше 49 балів — «незадовільно» / FX

Практика вважається зарахованою, якщо набрано 50 балів. Проект оцінюється в 100 балів. За виконання будь-якого пункту нараховується однакова кількість балів.

## **6.2. Перелік вимог до проекту технологічної практики**

### **6.2.1. Об’єктно-орієнтоване проектування та робота з даними**

В проекті мають бути використані або реалізовані перераховані нижче пункти.

1. Об’єктно-орієнтована проектування предметної області задачі.
2. Створення РОСО-класів.
3. Використання механізму Code First (Data First) для генерації таблиць БД.
4. Створення контексту даних.
5. Редагування таблиць та зв’язків між ними.
6. Написання запитів та рівні DAL (засобами LINQ, л-виразів, чи збережених процедур).
7. Реалізації CRUD – операцій.
8. Реалізація контролерів або обробників подій.

9. Реалізація в представлення або формах переліку карток об'єктів, детальних характеристик.

10. Доступ до даних із контролера або форми.

11. Редагування даних.

12. Валідація даних;

13. Зміна схеми бази даних, міграція. (додатково)

14. Адміністрування користувачів на основі ASPNET Membership чи ASPNET Identity з використанням ролей.

15. Використання конфігураційних файлів для задання постійних параметрів програми.

16. Використання Web-сервісів (додатково).

17. Використання маршрутизації (для Web-застосувань).

### **6.2.2. Дизайн застосування**

В проекті мають бути використані або реалізовані перераховані нижче елементи дизайну.

1. Наявність слайдерів або інших візуальних ефектів.

2. Адаптивний дизайн.

3. Використання Master Page або \_Layout та файлів стилів;

4. Наявність сторінок з категоріями об'єктів.

5. Наявність сторінок з переліком об'єктів та детальних характеристик об'єкта.

6. Наявність оберненого зв'язку.

7. Наявність авторизації користувачів.

8. Впорядкування, фільтрація та пошук.

9. Валідація даних.

10. Карта із зазначенням місця розташування об'єкту, GPS координати і дані повинні.

## **7. ПЕРЕЛІК НАВЧАЛЬНОЇ ЛІТЕРАТУРИ ТА ПОСИЛАНЬ НА РЕСУРСИ**

### **Література**

Programming Microsoft® ASP.NET 4, 2-е изд. — Redmond, Washington 98052-6399.: Microsoft Press, 2011. — 966, ISBN: 978-0-7356-4338-3

2. Разработка Web- приложений на Microsoft Visual Basic .NET и Microsoft Visual C# .NET. Учебный курс MCAD/MCSD/Пер. с англ. — М.: Издательско-торговый дом «Русская Редакция», 2003. — 704 с.: ил.

3. Мак-Дональд, Мэтью, Фримен, Адам, Шпушта, Марио. М15 Microsoft ASP.NET 4 с примерами на C# 2010 для профессионалов 4, 4-е изд.:Пер. с англ — М

:HYPERLINK

"<https://www.amazon.com/Tony->

N

б

б

б

h

Р 5. Палермо Джеффри и др. ASP.NET MVC 4 в действии. – М.: Manning, 2012.

И408 с. – ISBN 978-1-617-29041-1

РHYPERLINK

"<https://www.amazon.com/Adam->

Д

е

#### ё **Посилання**

В 1. Erik Reitan. Getting Started with ASP.NET 4.5 Web Forms and Visual Studio 2013, last updated September 8, 2014 – <https://www.asp.net/web-forms/overview/getting-started/getting-started-with-aspnet-45-web-forms/introduction-and-overview>

А 2. Rick Anderson. Getting Started with ASP.NET MVC 5, | last updated May 28, 2015 – <https://www.asp.net/mvc/overview/getting-started/introduction/getting-started>

Б 3. MVC recommended tutorials and articles By Rick Anderson | last updated May 22, 2015 – <https://www.asp.net/mvc/overview/getting-started/mvc-learning-sequence>

б 4. Rick Anderson, Tom Dykstra, Shayne Boyer. Getting started with ASP.NET Core MVC and Visual Studio, | last updated 14.10.2016 – <https://docs.asp.net/en/latest/tutorials/first-mvc-app/start-mvc.html>

б 5. Изучаем ASP.NET MVC 4 - <http://metanit.com/sharp/mvc/>

#### б **Відео**

М ASP.NET Core Deep Dive into MVC  
<https://channel9.msdn.com/Events/Build/2016/B812>

Н

Р-

/