

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«Київський політехнічний інститут імені Ігоря Сікорського»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра програмного забезпечення комп'ютерних систем

## **ТЕХНІЧНЕ ЗАВДАННЯ НА КУРСОВУ РОБОТУ**

*з дисципліни "Основи програмування"*

*на тему:*

***“Програмний додаток електронної бази  
навчальних курсів”***

Виконав: Жученко Андрій Сергійович  
Група: КП-02

Затверджено

---

2 семестр 2020/2021

Зміст

1.1. Найменування програми	3
1.2. Призначення і галузь застосування	3
<b>2. Вимоги до програми</b>	<b>4</b>
2.1. Вимоги до функціональних характеристик	4
2.2. Вимоги до надійності	5
2.2.1. Відмови через некоректні дії користувачів системи	5
<b>3. Умови експлуатації</b>	<b>6</b>
3.1. Кліматичні умови експлуатації	6
3.2. Вимоги до складу і параметрів технічних засобів	6
<b>4. Вимоги до програмної документації</b>	<b>6</b>
4.1. Попередній склад програмної документації	6
<b>5. Техніко-економічні показники</b>	<b>6</b>
5.1. Економічні переваги розробки	6
<b>6. Стадії і етапи розробки</b>	<b>7</b>
6.1. Стадії розробки	7
6.2. Етапи розробки	7
<b>7. Порядок контролю і приймання</b>	<b>7</b>
7.1. Види випробувань	7
7.2. Загальні вимоги до приймання роботи	8

# 1. Вступ

## 1.1. Найменування програми

Найменування програми: “Програмний додаток електронної бази навчальних курсів”.

## 1.2. Призначення і галузь застосування

Програма призначена для створення, керування вмістом бази даних, що містить такі дані:

1. Користувачів системи
2. Навчальних курсів

### 3. Лекцій

Програма надає консольний інтерфейс користувача для зручного керування записами в базі даних.

## Варіант теми

**n** - номер у списку групи.

Номер теми за варіантом: 4

Повна назва теми: “Програмний додаток електронної бази навчальних курсів”

Кожна тема має задані за варіантом сутності, а також обов’язкову сутність: **Користувач додатку**.

## Авторизація

Вимоги:

- Авторство дозволяє **користувачеві додатку** створювати сутності такого типу, модифікувати і видаляти такі сутності, що належать йому.
- Деякі варіанти мають розділення користувачів на просто користувачів і “модераторів”. Це задавати спеціальним значенням у типі **Користувач**.
- Якщо у вимогах не вказано інакше - користувачі мають доступ до зчитування даних сутностей.

## Варіант експорту даних і вимоги до імпорту

Варіант експорту даних: 0

Дані записувати у один XML файл.

Імпорт:

- Якщо при імпорті у БД додатку вже є дані з такими ідентифікаторами - ігнорувати імпортовані дані
- Якщо при імпорті у даних є ідентифікатори на неіснуючі записи у БД, створити записи з такими ідентифікаторами та додати до них спеціальну помітку “imported”.

## Генерація графіків та звіту

Генерація графіків:

- <https://swharden.com/scottplot/cookbooks/4.0.26/>

Генерація звіту:

- Створити .docx шаблон і програмно заповнювати його інформацією за варіантом.

## Електронна база навчальних курсів

Приклад: <https://www.coursera.org/>

Типи сутностей: **Курс, Лекція**

Зв'язки:

- Користувач-курс **many2many** - користувач може доєднатись до багатьох курсів
- Користувач-курс **one2many** - користувач є автором курсу
- Курс-лекція **one2many**

Авторизація:

- Користувач є автором курсу і зв'язаних з ним лекцій
- Користувач вирішує чи хоче від приєднатись чи від'єднатись від курсу

Додаткові функціональні вимоги:

- Автор курсу може відкривати чи закривати доступ до приєднання до його курсу
- Користувачі можуть переглядати лекції тільки тих курсів, до яких долучились

Варіант: 1

Експорт:

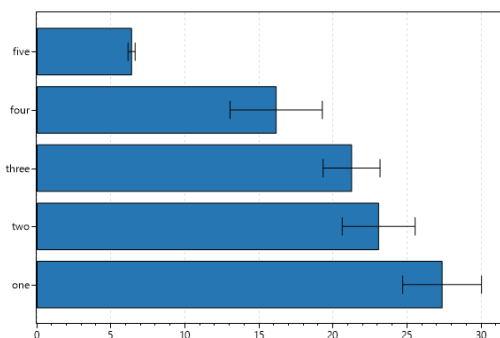
- Експортувати всі курси, що містять задане слово у назві і всі їх лекції.

Імпорт:

- Можна імпортувати всі експортовані курси з лекціями.

Генерація зображень:

- Згенерувати **horizontal bar** графік N курсів, з найбільшою кількістю підписників:



Генерація звіту:

- З інформацією про курси:
  - Кількість курсів
  - Середня кількість лекцій на курс
  - Середня кількість підписників на курс
  - Інформація про курс з найбільшою кількістю підписників
    - Автор
    - Назва
    - Кількість підписників

## 2. Вимоги до програми

Перед входом у систему користувач проходить аутентифікацію.

Система повинна забезпечувати одночасну роботу з одним сервером мінімум двох користувачів.

### 2.1. Вимоги до функціональних характеристик

Програма має задовольняти такі вимоги:

1. Наявний консольний інтерфейс користувача. Команди використані відповідно до їхнього основного призначення.
  - a. Наявно декілька станів інтерфейсу користувача.
  - b. Дотримується єдиний стиль для всіх команд
2. Наявні стани для створення і редагування даних.
  - a. Можна видаляти дані (з підтвердженням цієї дії).
  - b. Наявний стан редагування даних має мати вже заповнені поля із попередніми значеннями. Є можливість відмінити оновлення сутності.
3. Використання списків та таблиць для показу колекцій даних.
  - a. Дані в таблиці не є лише відображенням однієї таблиці БД, а містять склеєні чи змінені дані.
  - b. Дані у списках та таблицях пагінуються, доступний пошук та фільтрація їх записів.
4. Всі вхідні дані від користувача та з файлів перевіряються, помилки обробляються, користувачу відображається інформація про помилки.
5. Програма має можливість завантаження зображень з файлової системи, їх збереження у сховище даних та їх показ у інтерфейсі користувача.
6. Програма має можливість експорту-імпорту даних у форматі даних XML. Є можливість зберігання даних у файлах, та передача даних до інших процесів через протокол TCP.
7. Програма взаємодіє з реляційною базою даних. SQL запити до бази даних не конкатенуються, а їх виконання винесено у спеціальний окремий модуль (вид репозиторія).
  - a. База даних містить мінімум 3 таблиці. Таблиці в БД приведені до 1-ї та 2-ї нормальних форм.

- b. Між таблицями бази даних є one-to-many та/або many-to-many зв'язки
- 8. Наявна реєстрація користувачів у системі. Паролі користувачів не зберігаються у відкритому виді, а хешуються обраним алгоритмом хешування рядків.
  - a. Наявний стан аутентифікації. Пароль користувача при введенні прихований. Цей стан має сповіщення про некоректність введених користувачем даних.
  - b. Програма авторизує доступу до даних по користувачах. У базі є дані, що належать конкретному користувачу (наприклад, авторство сутностей).
- 9. Система дозволяє генерувати зображення та файлові звіти на основі даних, що зберігаються у БД.
- 10. Використання окремих потоків виконання для паралельних фонових дій.
- 11. Використання клієнт-серверної архітектури програм та TCP сокетів (розділення системи на клієнт і сервер з передачею даних у одному із стандартних форматів даних)
- 12. Частина проекту винесена у власні бібліотеки класів. Винесення саме цього коду обгрунтоване.
- 13. Наявний проект з модульними тестами функціональності системи.

## 2.2. Вимоги до надійності

### 2.2.1. Відмови через некоректні дії користувачів системи

Відмови програми внаслідок некоректних дій користувача при взаємодії з програмою через командний інтерфейс неприпустимі.

## 3. Умови експлуатації

### 3.1. Кліматичні умови експлуатації

Кліматичні умови експлуатації, при яких повинні забезпечуватися задані характеристики, повинні задовольняти

вимогам, що пред'являються до технічних засобів в частині умов їх експлуатації.

### 3.2. Вимоги до складу і параметрів технічних засобів

До складу технічних засобів повинен входити персональний комп'ютер, що включає в себе:

1. Оперативну пам'ять об'ємом 1 Гігабайт, не менше.
2. Операційну систему Windows 10 або Ubuntu Linux версії 18.04 чи вище.

## 4. Вимоги до програмної документації

### 4.1. Попередній склад програмної документації

Склад програмної документації повинен містити:

1. Технічне завдання
2. Звіт виконання

## 5. Техніко-економічні показники

### 5.1. Економічні переваги розробки

Орієнтовна економічна ефективність не розраховується. Аналогія не проводиться зважаючи на унікальності пропонованих вимог до розробки.

## 6. Стадії і етапи розробки

### 6.1. Стадії розробки

Розробка повинна бути проведена в три стадії:

1. Розробка технічного завдання
2. Робоче проектування
3. Захист курсової роботи

## 6.2. Етапи розробки

На стадії розробки технічного завдання повинен бути виконаний етап розробки і затвердження даного технічного завдання.

На стадії робочого проектування повинні бути виконані перелічені етапи робіт:

1. Розробка програми
2. Розробка програмної документації (звіту)
3. Тестування програми.

На стадії захисту курсової роботи повинні бути виконані етапи підготовки результатів виконання і захист результатів з демонстрацією їх Викладачу.

## 7. Порядок контролю і приймання

### 7.1. Види випробувань

Приймально-здавальні випробування повинні проводитись на об'єкті Студента або Викладача в обумовлені терміни.

Приймально-здавальні випробування включають збірку розробленої програми із проекту репозиторія Студента та демонстрації всієї розробленої функціональності.

### 7.2. Загальні вимоги до приймання роботи

У випадку виникнення труднощів в реалізації функцій програми, ТЗ може бути відкориговане. Пропозиції щодо зміни і доповнення ТЗ можуть вносити як Студент так і Викладач.