

Завдання для іспиту, виберіть одне з запропонованих, виконайте усі умови завдання, як результат надішліть посилання на Github репозиторій.

## Перший варіант:

Створити додаток для обліку продажу автомобілів в автосалоні. Основне завдання — враховувати процес продажу автомобілів, фіксувати співробітника, який провів операцію, підрахувати прибуток.

**Додаток має зберігати таку інформацію:**

**Співробітник:**

- П.І.Б.
- Посада.
- Контактний телефон.
- Email.

**Автомобіль:**

- Назва виробника.
- Рік випуску.
- Модель.
- Собівартість.
- Потенційна ціна продажу.

**Продажі:**

- Співробітник.
- Автомобіль.
- Дата продажу.
- Реальна ціна продажу.

**Необхідно зберігати таку функціональність:**

- Додавання та видалення інформації про співробітників.
- Додавання та видалення інформації про автомобілі.
- Додавання та видалення інформації про продаж.

- Звіти. Дані можуть виводитися на екран або файл, залежно від вибору користувача:
  - Повна інформація про співробітників фірми.
  - Повна інформація про автомобілі.
  - Повна інформація про продаж.
  - Усі продажі за певну дату.
  - Усі продажі за певний період часу.
  - Усі продажі певного співробітника.
  - Назва автомобіля, що найбільше продається за вказаний період часу.
  - Інформація про найуспішнішого продавця за вказаний період часу.
  - Сумарний прибуток за вказаний період часу.
- Збереження даних у файл.
- Завантаження даних з файлу.

Правильне використання патернів проєктування, SOLID-принципів, механізмів тестування під час реалізації завдання дозволить отримати більш високу оцінку.

## Другий варіант:

Створіть додаток для обліку продажу книг у книгарні. Основне завдання — враховувати процес продажу книг, фіксувати співробітника, який провів операцію, підрахувати прибуток.

**Додаток має зберігати таку інформацію:**

**Співробітник:**

- П.І.Б.
- Посада.
- Контактний телефон.
- Email.

**Книга:**

- Назва книги.
- Рік видання.
- Автор.

- Жанр.
- Собівартість.
- Потенційна ціна продажу.

#### **Продажі:**

- Співробітник.
- Книга.
- Дата продажу.
- Реальна ціна продажу.

#### **Необхідно зберігати таку функціональність:**

- Додавання та видалення інформації про співробітників.
- Додавання та видалення інформації про книги.
- Додавання та видалення інформації про продаж.
- Звіти. Дані можуть виводитися на екран або файл, залежно від вибору користувача:
  - Повна інформація про співробітників книгарні.
  - Повна інформація про книги.
  - Повна інформація про продаж.
  - Усі продажі за певну дату.
  - Усі продажі за певний період часу.
  - Усі продажі певного співробітника.
  - Назва книги, яка стала бестселером за вказаний період часу.
  - Інформація про найуспішнішого продавця за вказаний період часу.
  - Сумарний прибуток за вказаний період часу.
  - Найпопулярніший автор за вказаний період часу.
  - Найпопулярніший жанр за вказаний період часу.
- Збереження даних у файл.
- Завантаження даних з файлу.

Правильне використання патернів проєктування, SOLID-принципів, механізмів тестування під час реалізації завдання дозволить отримати більш високу оцінку.

## **Третій варіант:**

Реалізуйте додаток **«Облік витрат»**. Основне завдання додатка — зберігати інформацію про витрати користувача.

**Інтерфейс додатка має надавати такі можливості:**

- Додавання категорій витрат.
- Видалення категорій витрат.
- Додавання певної витрати. Потрібно вказувати таку інформацію:
  - Назва витрати.
  - Дата.
  - Категорія.
- Збереження витрат у файл.
- Завантаження витрат із файлу.
- Створення звітів за такими параметрами:
  - За датою.
  - За назвою.
  - За категорією.
  - Відображення максимальної витрати у кожній категорії.
  - Відображення максимальної витрати у вказаному періоді часу.
  - Відображення мінімальної витрати у кожній категорії.
  - Відображення мінімальної витрати у вказаному періоді часу.

Правильне використання патернів проєктування, SOLID-принципів, механізмів тестування під час реалізації завдання дозволить отримати більш високу оцінку.

## Четвертий варіант:

Створіть додаток **«Тлумачні Словники»**. Основне завдання проєкту — зберігати словники різними мовами і дозволяти користувачу знаходити тлумачення потрібного слова чи фрази.

**Інтерфейс додатка має надавати такі можливості:**

- Створювати словник. Під час створення необхідно вказати мову словника. Наприклад, українська або англійська.

- Додавати слово і його тлумачення до існуючого словника. Оскільки слово може мати декілька тлумачень, необхідно підтримувати можливість створення кількох варіантів.
- Замінювати слово або його тлумачення у словнику.
- Видаляти слово або тлумачення. Якщо слово видаляється, усі його тлумачення видаляються разом із ним. Не можна видалити тлумачення слова, якщо це останній варіант тлумачення.
- Шукати тлумачення слова.
- Словники мають зберігатися у файлах.
- Слово та варіанти його тлумачень можна експортувати до окремого файлу результату.
- Під час старту програми необхідно відображати меню для роботи з програмою. Якщо вибір пункту меню відкриває підменю, тоді в ньому потрібно передбачити можливість повернення до попереднього меню.

Правильне використання патернів проєктування, SOLID-принципів, механізмів тестування під час реалізації завдання дозволить отримати більш високу оцінку.

## П'ятий варіант:

Створіть додаток **«Вікторина»**. Основне завдання проєкту — надати користувачу можливість перевірити свої знання у різних галузях.

**Інтерфейс додатка має надавати такі можливості:**

- Під час старту додатка користувач вводить логін і пароль для входу. Якщо користувач не зареєстрований, він має пройти процес реєстрації.
- При реєстрації потрібно вказати:
  - Логін (не можна зареєструвати вже існуючий логін).
  - Пароль.
  - Дату народження.
- Після входу в систему користувач може:
  - Розпочати нову вікторину.
  - Переглянути свої результати з пройдених вікторин.
  - Переглянути Топ-20 з певної вікторини.

- Змінити налаштування. Можна змінювати пароль та дату народження.
- Вихід.
- Для початку нової вікторини користувач повинен обрати розділ знань вікторини. Наприклад, "Історія", "Географія", "Біологія" і т. д. Також потрібно передбачити змішану вікторину, коли питання обиратимуться з різних вікторин за випадковим принципом.
- Кожна вікторина складається із двадцяти питань. Кожне питання може мати один або декілька правильних варіантів відповідей. Якщо питання передбачає декілька правильних відповідей, а користувач вказав не все, питання не зараховується.
- Після завершення вікторини користувач отримує кількість правильних відповідей, а також отримує своє місце в таблиці результатів гравців вікторини.

Необхідно також розробити утиліту для створення та редагування вікторин та їх питань. Цей додаток має передбачати вхід за логіном і паролем.

Правильне використання патернів проектування, SOLID-принципів, механізмів тестування під час реалізації завдання дозволить отримати більш високу оцінку.

## Шостий варіант:

Створити додаток **«Житловий менеджмент»**. Основне завдання проєкту — надати користувачу можливість зберігати інформацію про мешканців будинку.

**Інтерфейс додатка має надавати такі можливості:**

- Додавання мешканців будинку.
- Видалення мешканців будинку.
- Додавання квартир (обов'язкове додавання інформації про поверх).
- Видалення квартир.
- Закріплення мешканців за квартирою.
- Відкріплення мешканців від квартири.
- Збереження інформації у файл.
- Завантаження інформації з файлу.
- Створення звітів за такими параметрами:

- Відображення повного списку мешканців.
- Відображення повного переліку квартир.
- Відображення інформації про певну квартиру.
- Відображення інформації про квартири на певному поверсі.
- Відображення інформації про квартири одного типу (наприклад, відобразити усі однокімнатні квартири).

Правильне використання патернів проєктування, SOLID-принципів, механізмів тестування під час реалізації завдання дозволить отримати більш високу оцінку.