**Черкаський національний університет імені Богдана Хмельницького**

**КАФЕДРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ АВТОМАТИЗОВАНИХ СИСТЕМ**

КУРСОВА РОБОТА

***з дисципліни “Об’єктно-орієнтоване програмування”***

***НА ТЕМУ «Інформаційна система підтримки продажу транспортних засобів»***

Студента 2 курсу, групи КС-202

напряму підготовки «Програмна інженерія» спеціальності «121 - Інженерія програмного забезпечення»

Невмитого Олега Миколайовича

|  |  |
| --- | --- |
| Керівник |  |
|  |  |
|  | **(посада, вчене звання, науковий ступінь, прізвище та ініціали )** |

Національна шкала:

Кількість балів: Оцінка: ECTS

Члени комісії

|  |  |  |
| --- | --- | --- |
| **(підпис)** |  | **(прізвище та ініціали)** |
|  |  |  |
| **(підпис)** |  | **(прізвище та ініціали)** |
|  |  |  |
| **(підпис)** |  | **(прізвище та ініціали)** |
|  |  |  |

**м. Черкаси – 2022 рік**

Зміст

[**Вступ** 3](#_Toc104990757)

[**Розділ 1. Огляд алгоритмів роботи та інструментарію для реалізації програми.** 5](#_Toc104990758)

[**1.1.** **Опис алгоритму роботи програми** 5](#_Toc104990759)

[**1.2.** **Вибір структур даних для реалізації даного програмного продукту** 5](#_Toc104990760)

[**1.3.** **Вибір графічної бібліотеки** 6](#_Toc104990761)

[**1.4.** **Висновок до першого розділу** 6](#_Toc104990762)

[**Джерела** 8](#_Toc104990763)

**Вступ**

**Мета** цієї курсової роботи є створення додатку для системи підтримки продажу транспортних засобів, з використанням трьох основних принципів об’єктно-орієнтованого програмування (ООП), а саме:

1. Інкапсуляція
2. Наслідування
3. Поліморфізм

Об’єктно-орієнтоване програмування (ООП) - одна з перших парадигм в програмуванні. Вона вирішує головну проблему – що робити зі складною предметною областю і складним кодом. Суть цієї парадигми сприймати всю предметну область у вигляді об’єктів, які якимсь чином взаємодіють один з одним. Оскільки людському мозку легше мислити об’єктами, ми автоматично розуміємо, що і у якого об’єкта має бути. Людині легко зрозуміти, де розташувати ті чи інші методи в коді. Тому з ООП ми отримуємо найбільш зрозумілу структуру програми. Основні принципи об’єктно орієнтованого програмування включаються в себе інкапсуляцію, наслідування та поліморфізм. [1]

Інкапсуляція - це коли кожен об’єкт зберігає свій стан всередині класу приватним. Інші об’єкти не мають прямого доступу до цього стану. Замість цього вони можуть викликати лише список відкритих функцій, які називаються методами. Отже, об’єкт керує своїм власним станом за допомогою методів — і жоден інший клас не може використати його, якщо це явно не дозволено. Якщо ви хочете зв’язатися з об’єктом, ви повинні використовувати надані методи. Але ви не можете змінити стан. [2]

Наслідування - це здатність набувати властивостей існуючих класів і створювати нові. Воно дозволяє повторно використовувати код без необхідності переписувати його в іншому класі. Однією з найкращих особливостей успадкування є можливість скорочувати код. Тобто ви можете використовувати цей принцип, щоб успадкувати код від іншого класу та повторно використовувати його в новому класі. [3]

Поліморфізм – це здатність однієї функції виконувати різні способи. Іншими словами, це стосується здатності об’єкта приймати більше ніж одну форму. Поліморфізм можна застосувати двома простими способами.

1. Перевантаження методу
2. Перевизначення методу

Перевантаження методу – це коли клас має кілька методів з однаковими іменами, але різним набором параметрів. Перевантаження можна використати лише якщо ваші методи задовольняють будь-які з наступних правил:

1. Мають різні параметри
2. Мать різний тип повертаючих даних. [4]

Перевизначення методу – це коли, дочірні класи реалізують при наслідуванні ті самі методи, що і батьківський, але логіка його роботи відрізняється він інших його нащадків, чи самого батьківського класу.

**Завдання:** написати систему підтримки продажу транспортних засобів, за допомогою якої можна буде підібрати собі транспортний засіб. Транспортні засоби сортуються за цілями використання (перевезення вантажів різного виду, перевезення людей різної кількості, тощо) та видами палива, що вони використовують (бензин, дизель, електрика). Також використати основні принципи ООП (інкапсуляція, наслідування, поліморфізм).

Для досягнення мети цієї курсової роботи необхідно вирішити наступні завдання:

1. Вибрати оптимальний алгоритм створення програмного продукту;
2. Сформувати загальні правила та алгоритм реалізації з використанням принципів ООП;
3. Побудувати блок-схему загального алгоритму;
4. Побудувати діаграму класів програми;
5. Реалізувати програмний продукт;
6. Зробити висновки по роботі.

# **Розділ 1. Огляд алгоритмів роботи та інструментарію для реалізації програми.**

Відповідно до поставленої мети курсової роботи, для її реалізації було обрано об’єктно орієнтовану мову програмування C# із зручним синтаксисом, та підтримкою бібліотеки WPF для створення десктопних додатків з користувацьким інтерфейсом, що значно спрощує реалізацію потрібних функцій. Розробка буде проводитися в рідному для C# інтегрованому середовищі розробки - Visual Studio 2022.

## **Опис алгоритму роботи програми**

Під час запуску програми, при завантажені вікна програма спробує отримати дані з файлу JSON, де лежать серіалізовані об’єкти транспортних засобів, та завантажить ці дані у таблицю на головній сторінці. Після успішного запуску програми, вона матиме такі функції:

1. Сортувати отримані дані в таблиці по категоріям (тип транспортного засобу, тип пального та станом справності)
2. Отримати інформацію про конкретний транспортний засіб
3. Додати в список новий транспортний засіб
4. Додати нові типи пального
   1. **Вибір структур даних для реалізації даного програмного продукту**

В реалізації потрібних нам функцій потрібно зберігати списки транспортних засобів та видів пального, оскільки дані будуть змінюватися динамічно, ми використаємо стандартну структуру даних – список (List<T>). Для видалення зі списку об’єкта, ми реалізуємо в ньому інтерфей IEquatable і перевантажимо метод Equals, щоб при використанні методу Remove, воно шукало об’єкт по унікальному ідентифікаторі.

Для додаткових можливих функцій, таких як список делегатів ми використаємо звичайний масив (Array). Зберігати такі дані як стан функціювання та тип вантажу, де стани повинні бути статичними і не змінюватися будемо в перерахуваннях (Enum).

* 1. **Вибір графічної бібліотеки**

Графічний інтерфейс для даного програмного продукту буде зроблений за допомогою Windows Presentation Foundation (WPF). Це фреймворк для розробки користувацьких інтерфейсів, з використанням мови розмітки XAML.

* 1. **Висновок до першого розділу**

В даному розділі було описано основний алгоритм роботи програмного продукту, обрано мову програмування (С#), структури даних (List<T>, Array, Enum) та фреймворк для написання графічного інтерфейску (WPF).

# **Розділ 2. Проектування інформаційної системи підтримки продажу транспортних засобів**

* 1. **Функції для реалізації програмного продукту**

Використання програми буде проходити наступним чином:

в програмі при старті ініціалізуються дані про добавлені транспортні засоби, які зберігаються в JSON файлі. Користувацький інтерфейс має три сторінки, та бокову панель для того, щоб переключатися між ними.

Перша сторінка, тут знаходяться фільтри, по яким буде вибір які саме транспортні засоби показувати та сама таблиця з ними. В рядку таблиці будуть такі дані як назва, модель, ціна, та кількість сидячих місць у цьому транспортному засобі. Також кнопка “Get Info”, при натисканні на яку буде відкрите додаткове вікно з повною інформацією про вибраний транспортний засіб.

Друга і третя сторінки для запису нових транспортних засобів та запису нових видів палива відповідно, після чого програма збереже добавлені дані в файл JSON, за допомогою серіалізації.

Для збереження даних будуть написані DTO (Data Transfer Object), та адаптери, які будуть конвертувати дані з моделей в dto, та навпаки. Це потрібно для того, щоб можно було правильно серіалізувати дані у JSON формат.

* 1. **Узагальнена блок схема програми**

**Джерела**

1. ООП - <https://foxminded.ua/ua/shho-take-oop-ob-iektno-oriientovane-programuvannja>
2. Інкапсуляція - <https://www.freecodecamp.org/news/object-oriented-programming-concepts-21bb035f7260>
3. Наслідування - <https://www.partech.nl/en/publications/2020/10/basic-principles-of-object-oriented-programming#>
4. Поліморфізм - [https://www.partech.nl/en/publications/2020/10/basic-principles-of-object-oriented-programming#](https://www.partech.nl/en/publications/2020/10/basic-principles-of-object-oriented-programming)
5. List - <https://docs.microsoft.com/en-us/dotnet/api/system.collections.generic.list-1?view=net-6.0>