**Черкаський національний університет імені Богдана Хмельницького**

**КАФЕДРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ АВТОМАТИЗОВАНИХ СИСТЕМ**

КУРСОВА РОБОТА

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

***НА ТЕМУ «Програма супроводу роботи прокатного відділу мультимедійних матеріалів»***

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](#_Toc104848526)

[**Джерела** 4](#_Toc104848527)

**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.**

**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#>