Корпоративний стандарт

**ЗАГАЛЬНІ ВИМОГИ ДО ІНТЕРФЕЙСУ КОРИСТУВАЧА**

Дата введення 29-03-2021

Збараж - 2023

Оглавление

[**1. Розробка стандарту кодування 5**](#_Toc89334921)

[**1. Pascal casing 5**](#_Toc89334922)

[**3. Суфікси та префікси 5**](#_Toc89334923)

[**4. Абревіатури 5**](#_Toc89334924)

[**5. Найменування способів 6**](#_Toc89334925)

[**6. Змінні, поля та властивості 6**](#_Toc89334926)

[**7. Додаткові поради 6**](#_Toc89334927)

[**8. Об'єм коду 6**](#_Toc89334928)

[**9. Додаткові вимоги 6**](#_Toc89334929)

[**10. Приклад оформлення фрагменту коду мовою програмування С# 7**](#_Toc89334930)

[**2. Розробка стандарту інтерфейсу 7**](#_Toc89334931)

[**2.2.1 Додавання продукту до корзини 9**](#_Toc89334932)

[**2.2.2 Зміна продукту до корзини 10**](#_Toc89334933)

[**2.2.3 Видалення продукту з корзини 10**](#_Toc89334934)

[**2.2.4 Закінчення покупки 10**](#_Toc89334935)

[**2.2 Використання адміністратором 11**](#_Toc89334936)

[**2.2.1 Додавання нового товару 11**](#_Toc89334937)

[**2.2.2 Змінна існуючого товару 11**](#_Toc89334938)

[**2.2.3 Видалення товару 12**](#_Toc89334939)

[**2.2.4 Дізнатися історію зміни цин. 12**](#_Toc89334940)

[**2.2.5 Створення нової поставки. 12**](#_Toc89334941)

[**2.2.6 Змінна існуючої поставки 14**](#_Toc89334942)

[**2.2.7 Видалення поставки 14**](#_Toc89334943)

[**2.2.8 Створення звиту 14**](#_Toc89334944)

[**2.2.9 Закінчення сеансу 14**](#_Toc89334945)

[**3. Розробка технології зборки версії 15**](#_Toc89334946)

# **Розробка стандарту кодування**

Стандарт оформлення коду приймається та використовується групою розробників програмного забезпечення для єдиного стилю оформлення коду, над яким йде спільна праця. Стандарти кодування будуються так, щоб шляхом визначеного візуального оформлення елементів програми збільшити інформативність коду для людини.

Метою прийняття та використання стандарту є спрощення прийняття програмного коду людиною, мінімізація навантаження на пам’ять та зір при читанні програми.

Для написання програми «Рука допомоги» ми обрали мову програмування C#, так склали такі вимоги до написання коду:

1. **Pascal casing**

Описуються імена:   
• всіх визначень типів, у тому числі класів користувача, перерахувань, подій, делегатів і структур;   
• значення перерахувань;   
• readonly полів і констант;   
• інтерфейсів;   
• методів;   
• просторів імен (namespace);   
• властивостей;   
• громадських полів;

1. **Camel casing**  
   Описуються імена:   
   • локальних змінних;   
   • аргументів методів;   
   • захищених (protected) полів.
2. **Суфікси та префікси**  
   Застосовуються наступні суфікси та префікси:   
   • імена власних класів винятків завжди закінчуються суфіксом “Exception”;   
   • імена інтерфейсів завжди починаються з префіксу I;   
   • імена атрибутів користувача завжди закінчуються суфіксом «Attribute»;   
   • імена делегатів обробників подій завжди закінчуються суфіксом EventHandler, імена класів-спадкоємців від EventArgs завжди закінчуються суфіксом EventArgs.
3. **Абревіатури**

При використанні абревіатур в іменах капіталізації підлягають абревіатури з двома символами, в інших абревіатурах необхідно призводити до верхнього регістру тільки перший символ.

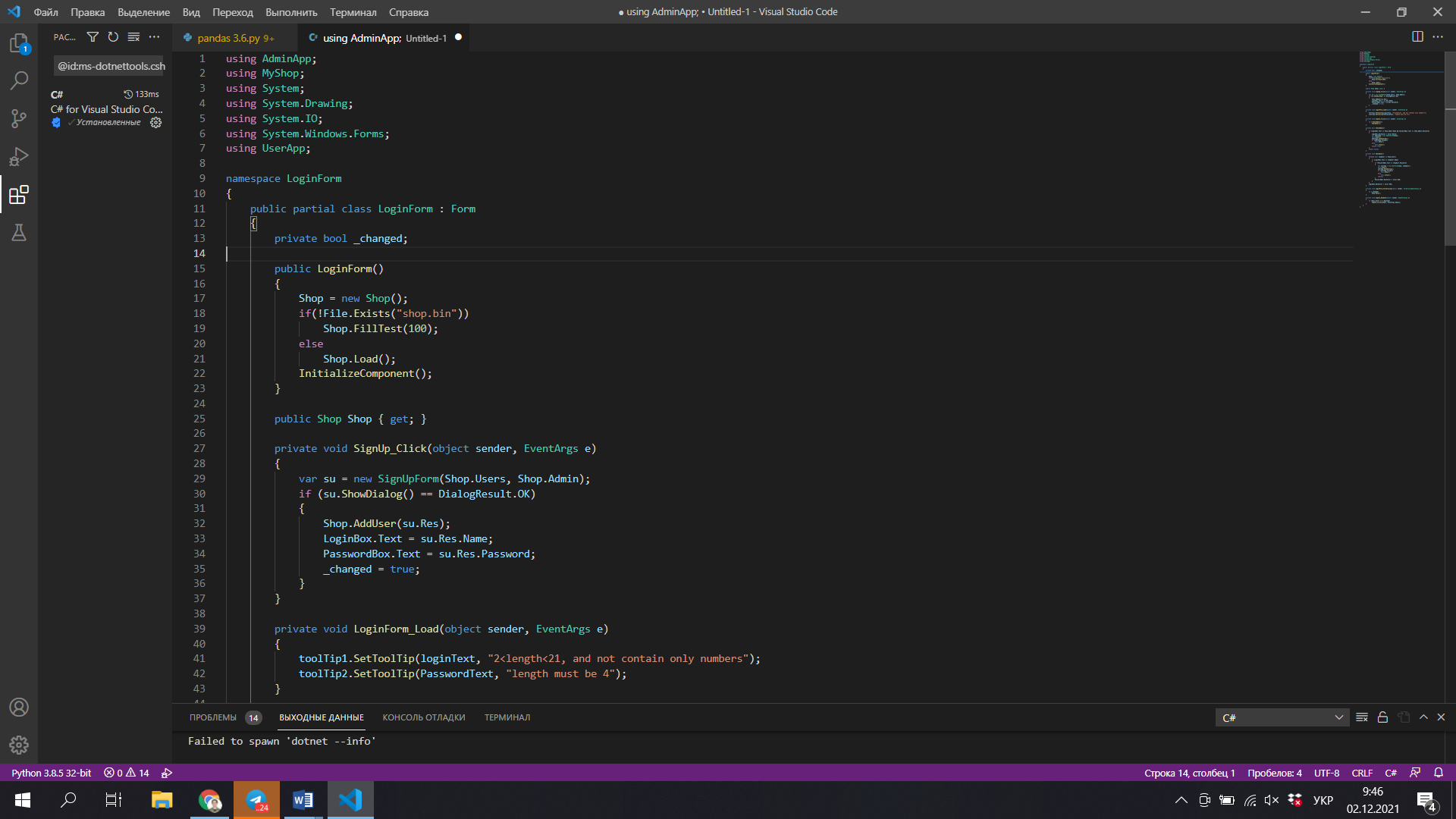
1. **Найменування способів**

Використовували конструкцію дієслово-об'єкт для іменування методів .  
У окремому випадку, для методів, які повертають значення, використовували в парі дієслово-об'єкт для дієслова «Get», а для об'єкта – опис значення, що повертається.

1. **Змінні, поля та властивості**  
   • При іменуванні змінних уникали використання скорочених варіантів на зразок I і t , використовували index та temp . Не використовували угорську нотацію або використовували її лише для закритих членів. Та не скорочували слова, використовували number , а не num .   
   • Для імен елементів керування вказували префікси, які описують тип елемента. Наприклад: txtSample. Ця рекомендація поширюється на локальні змінні складних типів.  
   • не використовували публічних чи захищених полів, натомість використовувалие властивості;   
   • використовували автоматичні властивості;   
   • завжди вказували модифікатор private, навіть якщо дозволено його опускати;   
   • завжди ініціалізували змінні, навіть коли існує автоматична ініціалізація.
2. **Додаткові поради**  
   • використовували порожній рядок між логічними секціями у вихідному файлі, класі, методі;   
   • використовували проміжну змінну для передачі bool-значення результату функції умовний вираз if ;
3. **Об'єм коду**  
   • старалися уникати файлів із більш ніж 500 рядками коду та методів із більш .ніж 200 рядками коду;   
   • уникали методів з більш ніж 5 аргументами, використовуйте структури передачі великої кількості параметрів;   
   • один рядок коду не має довжину понад 120 символів.
4. **Додаткові вимоги**

• розташовували фігурні дужки, що відкривають і закривають, на новому рядку;   
• використовували фігурні дужки для виразу if, навіть коли вираз входить лише один рядок коду;

Для імен змінних та інших ідентифікаторів ми використовували 3 типи регістрів символів: верхній, нижній і титульний в залежності від змісту тексту, який буде виводитись. Не використовували угорську нотацію, «верблюжий», «верблюжий» з малої букви та знаку підкреслення для розділу слів.

1. **Приклад оформлення фрагменту коду мовою програмування С#** 

# **Розробка стандарту інтерфейсу**

**Вхід в систему**

При успішному запуску програми наступним на екрані з’явиться вікно запрошення для входу в систему, яке містить:

* заголовок вікна,
* поле даних для введення імені з написом «Login»;
* поле даних для введення пароля з написом «Password»
* кнопки підтвердження «Вхід» та «Реєстрація»

Написи полів введення даних розташовуються зліва від них.

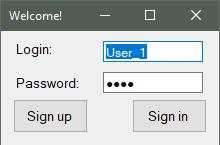
Після запуску програми відкривається вікно входу (рис. 3.1).

Рисунок 2.1 – Вікно входу в аккаунт

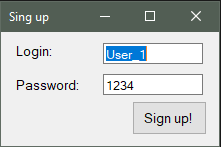
Бажаючи створити новий обліковий запис, користувач натиска “Sign up”.

Рисунок 2.2 – Вікно реєстрації

Користувач вводить свої дані. Якщо дані введені правильно, користувача повертає до форми входу.

Увійшовши до свого аккаунта користувач потрапляє до головної сторінки (рис. 2.2.1).

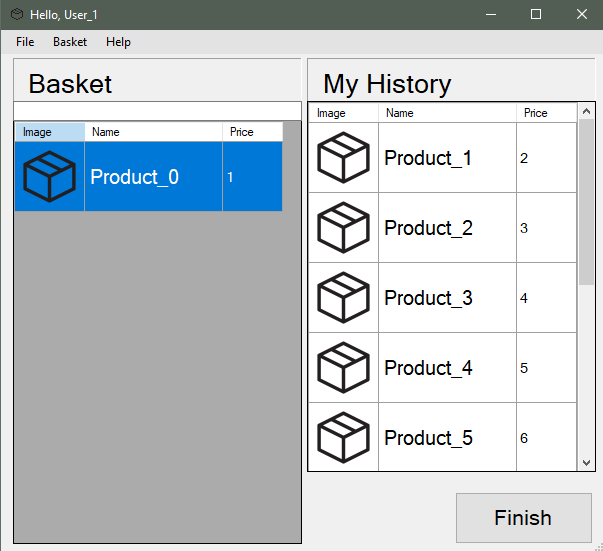


Рисунок 2.2.1 – Головне вікно користувача

2.2.1 Додавання продукту до корзини

У меню необхідно обрати “Basket”, затим “Add”, після чого відкриваеться вікно додавання необхідних речей для замовлення (рис. 2.2.2).

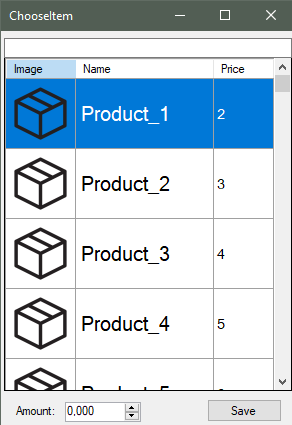
Обравши необхідний товар та його кількість, натискаймо кнопку «Save». Після чого знов відкритеся головне вікно (рис. 2.2.1), де з’явиться обраний товар.

Рисунок 2.2.2 – Вибір продукту

2.2.2 Зміна продукту до корзини

За необхідністю є можливість змінити кількість придбання обраного товару, кликнувши на “Edit” в меню “Basket”. Після чого обрати бажану кількість.

2.2.3 Видалення продукту з корзини

Також можливо видалити товар, обравши “Delete” в меню “Basket”. Для попередження випадкового дотику необхідно буде підтвердити свій вибір.

2.2.4 Закінчення замовлення

Щоб завершити процес замовлення, необхідно обрати “Finish”, и після підтвердження потребуючий отримує список необхідних товарів.

Щоб повернутися до форми авторизації(рис 3.1) необхідно натиснути кнопку «To sign in» в меню “File”.

Щоб вийти з програми, натисніть кнопку «To Desktop» в меню “File”.

* 1. Використання адміністратором

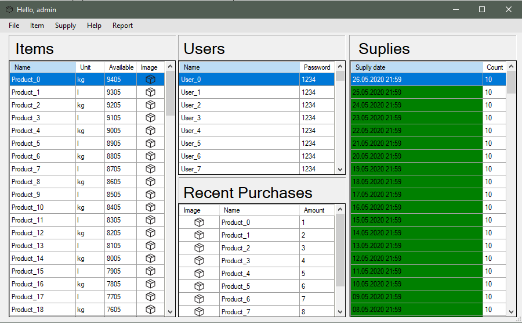
Увійшовши до свого аккаунта користувач потрапляє до адмінпанели (рис. 2.2.1).

Рисунок 2.2.1 – Адмінпанель

2.2.1 Додавання нового товару

Щоб додати новий товар до складу необхідно натикнути кнопку “New” у меню “Item”. У вікні, що відкрилось, заповнити данні про товар: Назва, одиниці вимірювання, зображення, та при необхідності доступна кількість. Після завершення заповнення натиснути кнопку “Save”

2.2.2 Змінна існуючого товару

Щоб виправити інформацію про товар необхідно натикнути кнопку “Edit” у меню “Item” або зробити подвійний клик на бажаний товар.

У вікні, що відкрилось, можливо виправити одиниці вимірювання, доступна кількість, зображення, та при умови, що товар не використовуваний у поточних поставках.

2.2.3 Видалення товару

Щоб видалити товар необхідно натикнути кнопку “Delete” у меню “Item”. Для попередження випадкового дотику необхідно буде підтвердити свій вибір. Видалити можливо лише той товар, що не використовуваний у поточних поставках.

2.2.4 Дізнатися історію кількості товарів.

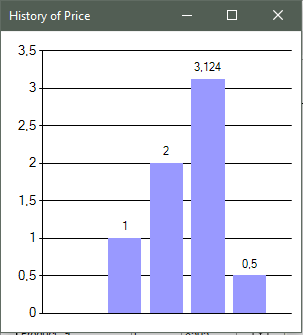
Щоб продивитися історію кількості товарів на обраний товар необхідно натикнути кнопку “Edit” у меню “Item” або зробити подвійний клик на бажаний товар, після чого обрати “Prices”.

Рисунок 2.2.2 - Вікно історії кількості товарів

2.2.5 Створення нової поставки.

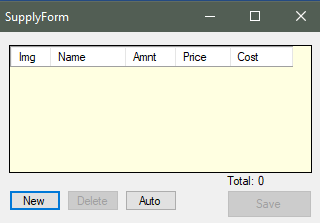
Щоб створити нову поставку необхідно натикнути кнопку “New” у меню “Supply”. З’явиться вікно обраних товарів(рис. 2.2.3) у поставці.

Рисунок 2.2.3 - Вікно поставки

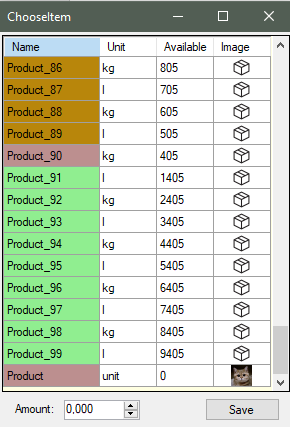
Щоб додати товари до поставки необхідно обрати “New”, у вікні, що відкрилось (рис. 2.2.4), обрати бажаний товар та його кількість. Для полегшення створення поставки товари доступна кількість яких менша за 500 одиниць відмічений червоним, вид 500 до 1000 одиниць – помаранчевим, інакше – зеленний.

Рисунок 2.2.4 - Вікно додавання товару до поставки

Щоб видалити обраний товар с поставки необхідно обрати “Delete”, у вікні обраних товарів (рис. 2.2.3).

Присутнє авто-замовлення, що дозволяє автоматично додати до замовлення усі товари, присутня кількість котрих менше за 500 одиниць.

Для підтвердження поставки неохайно обрати “Save”, у вікні обраних товарів (рис. 2.2.3).

Створене замовлення додається до списку всіх поставок у адмінпанелі ( рис. 2.2.1).

Ще не виконанні замовлення будуть відміченні червоним, виконані – зеленим.

2.2.6 Змінна існуючої поставки

Щоб виправити інформацію про поставку необхідно натикнути кнопку “Edit” у меню “Supply” або зробити подвійний клік на бажану поставку.

у вікні, що відкрилось, можливо додати/видалити товари у поставці.

Виправити можливо лише ту поставку, що обула створена не раніше двох годин тому.

2.2.7 Видалення поставки

Щоб видалити поставку необхідно натикнути кнопку “Delete” у меню “Supply”.

Для попередження випадкового дотику необхідно буде підтвердити свій вибір.

Видалити можливо лише ту поставку, що була створена не раніше двох годин тому.

2.2.8 Створення звіту

Для створення звиту необхідно натиснути кнопку “Report” у адмінпанелі (рис.2.2.1).

Підтвердивши шлях створення файлу, звіт буде створений під назвою “MyShop.txt” у заданому каталозі.

2.2.9 Закінчення сеансу

Щоб повернутися до форми авторизації(рис 2.1) необхідно натиснути кнопку «To sign in» в меню “File”.

Щоб вийти з програми, натисніть кнопку «To Desktop» в меню “File”.

Щоб зберегти корегування необхідно натиснути кнопку «Save» в меню “File”.

Щоб загрузити збережені данні магазину необхідно натиснути кнопку «Load» в меню “File”.

# **Розробка технології зборки версії**

**Інтегроване середовище розробки,** ІСР ( англ. *IDE, Integrated development environment або integrated debugging environment* ) - Система програмних засобів, використовувана програмістами для розробки програмного забезпечення (ПО).

Зазвичай середу розробки включає в себе:

* текстовий редактор
* компілятор та / або інтерпретатор
* засоби автоматизації збирання
* відладчик.

Для нашої програми “Рука допомоги” ми обрали Visual Studio Code - засіб для створення, редагування та зневадження сучасних вебзастосунків і програм для хмарних систем. Перевагами даного IDE є те, VSCode дозволяє обмінюватися модулями між програмістами, за допомогою того, що VSCode підтримує роботу з Git та Azure Pipelines. Також VSCode імпортує дані локальних або віддалених баз даних із підтримкою Azure Cosmos DB, MongoDB і т.д і керує ними.