# ТЕХНІЧНІ ХАРАКТЕРИСТИКИ

## Постановка задачі

### Характеристика об’єкта автоматизації

Оптова база - це комплексна торгівельна організація, яка спеціалізується на продажі оптових партій продуктів харчування, товарів для дому та особистої гігієни. Оптові бази мають розгалужену мережу підрозділів, які знаходяться в різних районах міста та регіонів.

Основні функції оптової бази включають продаж продуктів харчування, управління запасами та планування закупівель, керування продажами та клієнтською базою даних, ведення бухгалтерського обліку.

Для автоматизації діяльності оптової бази необхідне програмне забезпечення, яке дозволить ефективно управляти складським обліком, контролювати рух товарів, планувати закупівлі, керувати продажами та клієнтською базою даних, а також вести бухгалтерський облік. Важливим параметром є також прибутковість продуктів та обсяги продажів.

Для забезпечення ефективної автоматизації оптової бази, програмне забезпечення повинно бути інтегровано з програмами обліку та управління запасами, які вже використовуються в компанії. Це дозволить створити єдину інформаційну систему, яка забезпечить повну інтеграцію всіх підрозділів та ефективний контроль за управлінням компанією.

### 1.1.2 Вимоги до програми

Призначення програмного забезпечення та програмного продукту: розробка системи автоматизації управління та аналізу продажами в оптовій базі.

Перелік задач, які будуть вирішуватися в результаті використання розробленого програмного продукту:

* Облік товарів на складі;
* керування закупівлями товарів;
* ведення обліку продажів;
* взаємодія з клієнтами та програма лояльності;
* генерація звітів та аналіз даних.

Структура програмного продукту має наступний вигляд:

* Модуль перегляду та додавання продуктів;
* модуль замовлення товарів;
* модуль аналізу даних;
* модуль налаштування системи.

Опис функціонування програмного продукту та його частин:

* Модуль перегляду та додавання продуктів надає можливість переглянути список всіх продуктів зареєстрованих в системі, додавати нові продукти та редагувати уже існуючі;
* модуль замовлення товарів надає можливість закупляти товари та редагувати уже існуючі документи закупівель;
* модуль аналізу даних аналізує дані та генерує звіти для управління та оптимізації бізнес-процесів;
* модуль налаштування системи надає можливість створювати, редагувати та активовувати/деактивовувати облікові записи користувачів системи.

Користувачами програмного продукту можуть бути менеджер з продаж, що використовує програму на робочому комп’ютері під час взаємодії з клієнтами. Також програма може використовуватись адміністратором оптової бази, який може виконувати всі вище згадані задачі.

## Опис математичного методу рішення задачі

При розробці поставленої задачі було використано метод об’єктно-орієнтованого програмування.

Об’єктно-орієнтоване програмування (ООП) — одна з парадигм програмування, яка розглядає програму як множину взаємодіючих об'єктів. У ній використано декілька технологій від попередніх парадигм, включаючи успадкування, модульність, поліморфізм та інкапсуляцію. Об'єктно-орієнтоване програмування сягає своїм корінням до створення мови програмування Симула в 1960-тих роках, одночасно з посиленням [дискусій](http://ua-referat.com/%D0%94%D0%B8%D1%81%D0%BA%D1%83%D1%81%D1%8B) про кризу програмного забезпечення. Незважаючи на те, що ця [парадигма](http://ua-referat.com/%D0%9F%D0%B0%D1%80%D0%B0%D0%B4%D0%B8%D0%B3%D0%BC%D0%B0) з'явилась в 1960-тих роках, вона не мала широкого застосування до 1990-тих.

Об'єктно-орієнтовані програми можна розглядати як множину взаємодіючих об'єктів, на відміну від традиційних поглядів, коли програма розглядається як набір підпрограм, або як перелік інструкцій комп'ютеру. [Відповідно](http://ua-referat.com/%D0%92%D1%96%D0%B4%D0%BF%D0%BE%D0%B2%D1%96%D0%B4%D1%8C) до парадигми ООП, кожен об'єкт здатен отримувати повідомлення, обробляти дані, та надсилати повідомлення іншим об'єктам. Кожний об'єкт можна розглядати як незалежний [автомат](http://ua-referat.com/%D0%90%D0%B2%D1%82%D0%BE%D0%BC%D0%B0%D1%82) з окремим призначенням або відповідальністю.

Об'єктно-орієнтований підхід полягає в наступному наборі основних принципів:

* все є об'єктами;
* всі дії та розрахунки виконуються шляхом взаємодії (обміну даними) між об'єктами, при якій один об'єкт потребує, щоб інший об'єкт виконав деяку дію. Об'єкти взаємодіють, надсилаючи і отримуючи повідомлення. Повідомлення - це запит на виконання дії, доповнений набором аргументів, які можуть знадобитися при виконанні дії;
* кожен об'єкт має незалежну пам'ять, яка складається з інших об'єктів;
* кожен об'єкт є представником (екземпляром, примірником) класу, який виражає загальні властивості об'єктів;
* у класі задається поведінка (функціональність) об'єкта. Таким чином усі об'єкти, які є екземплярами одного класу, можуть виконувати одні й ті ж самі дії;
* класи організовані у єдину деревовидну структуру з загальним корінням, яка називається ієрархією [успадкування](http://uk.wikipedia.org/wiki/%D0%A3%D1%81%D0%BF%D0%B0%D0%B4%D0%BA%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_%28%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F%29). Пам'ять та поведінка, зв'язані з екземплярами деякого класу, автоматично доступні будь-якому класу, розташованому нижче в ієрархічному дереві.

Об’єктно-орієнтоване програмування засноване на трьох принципах:

* інкапсуляції;
* наслідуванні;
* поліморфізмі.

Спостережуване в об'єктах об'єднання даних і операцій в одне ціле було позначене [терміном](http://ua-referat.com/%D0%A2%D0%B5%D1%80%D0%BC%D1%96%D0%BD%D0%B8) інкапсуляція. Застосування інкапсуляції зробило об'єкти схожими на маленькі програмні модулі і забезпечило приховування їх внутрішньої структури. Для об'єктів з'явилося [поняття](http://ua-referat.com/%D0%9F%D0%BE%D0%BD%D1%8F%D1%82%D1%82%D1%8F) інтерфейсу, що значно підвищило їх надійність і цілісність.

Наслідування означає, що при створенні нового класу, немає необхідності в переписуванні заново всіх полів, методів і властивостей. Ви оголошуєте, що новий клас є нащадком (або дочірнім класом) наявного класу, так званим предком (або батьківським класом), і додаєте до нього нові поля, методи та властивості. Іншими словами, додається те, що потрібне для переходу від загального до приватного. [Процес](http://ua-referat.com/%D0%9F%D1%80%D0%BE%D1%86%D0%B5%D1%81) створення нових класів на основі інших класів називається наслідуванням. Нові класи мають як успадковані ознаки, так і нові.

Поліморфізм означає, що в похідних класах можна змінювати роботу методів, що вже існують в базовому класі. При цьому весь програмний код, що керує об'єктами батьківського класу, придатний для управління об'єктами дочірнього класу без будь-якої модифікації.

Також у програмі використано ієрархію класів для створення динамічного графічного інтерфейсу та взаємодії даних.

## Визначення структури вхідних даних

Вхідними даними називають такі змінні (параметри, дані), які надаються виконавцеві до початку його роботи над алгоритмом.

Застосування алгоритму до неприпустимих(або неперевірених) вхідних даних може призводити до того, що алгоритм ніколи не зупиниться, або потрапить в тупиковий стан (зависання), з якого не зможе продовжитися виконання процесу.

Таблиця 1.1 – Вхідна інформація

|  |  |  |  |
| --- | --- | --- | --- |
| №  з/п | Назва | Форма подання | Термін і частота використання |
| 1 | Товар | Масив оперативної інформації | Не обмежено |
| 2 | Працівники | Масив оперативної інформації | Не обмежено |
| 3 | Клієнт | Масив оперативної інформації | Не обмежено |
| 4 | Товарна накладна | Масив оперативної інформації | Не обмежено |
| 6 | Виробник | Масив нормативно-довідкової інформації | Не обмежено |
| 7 | Постачальник | Масив нормативно-довідкової інформації | Не обмежено |
| 8 | Кошик | Масив оперативної інформації | Не обмежено |
| 9 | Користувач | Масив нормативно-довідкової інформації | Не обмежено |
| 10 | Документ списання | Масив оперативної інформації | Не обмежено |

Вхідні дані зазвичай поділяють на два класи: довідкову та оперативну інформацію. Довідкові дані змінюються рідше, ніж оперативні, які можуть змінюватися навіть в режимі реального часу.

Вхідні дані представлені у вигляді таблиць, що зберігаються в базі данних.

Таблиця 1.2 – Структура вхідних даних таблиці «Товар»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № з/п | Ідентифікатор | Тип | Діапазон значень | Пояснення |
| 1 | 2 | 3 | 4 | 5 |
| 1. | Pr\_Id | int | Довге ціле | Унікальний ідентифікатор товару |
| 2. | Pr\_BarCode | varchar(20) |  | Штрих-код товару |
| 3. | Pr\_Description | varchar(max) |  | Опис товару |
| 4. | Pr\_Brand | int | Довге ціле | Ідентифікатор бренду товару |
| 5. | Pr\_Title | varchar(200) |  | Назва товару |
| 6. | Pr\_Category | int | Довге ціле | Ідентифікатор категорії товару |
| 7. | Pr\_PriceOpt | decimal(18,2) |  | Оптова ціна товару |
| 8. | Pr\_Price | decimal(18,2) |  | Роздрібна ціна товару |
| 9. | Pr\_Qty | varchar(20) |  | Одиниці виміру |
| 10. | Pr\_Reorder | int | Довге ціле | Критичний залишок |
| 12. | Pr\_Image | varbinary(MAX) |  | Зображення товару |

Таблиця 1.3 – Структура вхідних даних таблиці «Списання»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № з/п | Ідентифікатор | Тип | Діапазон значень | Пояснення |
| 1 | 2 | 3 | 4 | 5 |
| 1. | Adj\_Id | int | Довге ціле | Унікальний ідентифікатор |
| 2. | Adj\_Product | int | Довге ціле | Ідентифікатор товару |
| 3. | Adj\_Qty | int | Довге ціле | Кількість списаного товару |
| 4. | Adj\_Remarks | varchar | 0..50 | Додаткові зауваження до налаштування товару |
| 5. | Adj\_Date | datetime |  | Дата |
| 6. | Adj\_User | varchar | 0.50 | Користувач |

Таблиця 1.4 – Структура вхідних даних таблиці «Кошик»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № з/п | Ідентифікатор | Тип | Діапазон значень | Пояснення |
| 1 | 2 | 3 | 4 | 5 |
| 1. | Crt\_Id | int | Довге ціле | Унікальний ідентифікатор |
| 2. | Crt\_Transno | varchar | 0..50 | Номер транзакції |
| 3. | Crt\_Product | int | Довге ціле | Ідентифікатор товару |
| 4. | Crt\_Price | decimal |  | Ціна товару |
| 5. | Crt\_Qty | int | Довге ціле | Кількість товару |
| 6. | Crt\_Disc\_Percent | decimal |  | Відсоток знижки |
| 7. | Crt\_Discount | decimal |  | Сума знижки |
| 8. | Crt\_Total | decimal |  | Загальна сума |

Продовження таблиці 1.4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| 9. | Crt\_Date | datetime |  | Дата та час |
| 10. | Crt\_Status | int | Довге ціле | Статус |
| 11. | Crt\_Cashier | int | Довге ціле | Ідентифікатор касира |

Таблиця 1.5 – Структура вхідних даних таблиці «Категорія»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № з/п | Ідентифікатор | Тип | Діапазон значень | Пояснення |
| 1 | 2 | 3 | 4 | 5 |
| 1. | Catg\_Id | int | Довге ціле | Унікальний ідентифікатор |
| 2. | Catg\_Title | varchar | 0..50 | Назва категорії |
| 3. | Catg\_Description | text |  | Опис категорії |

Таблиця 1.6 – Структура вхідних даних таблиці «Клієнт»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № з/п | Ідентифікатор | Тип | Діапазон значень | Пояснення |
| 1 | 2 | 3 | 4 | 5 |
| 1. | Cl\_Id | int | Довге ціле | Унікальний ідентифікатор |
| 2. | Cl\_Name | varchar | 0..50 | ПІБ клієнта |
| 3. | Cl\_CardNumber | varchar | 10 | Номер картки клієнта |

Таблиця 1.7 – Структура вхідних даних таблиці «Працівник»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № з/п | Ідентифікатор | Тип | Діапазон значень | Пояснення |
| 1 | 2 | 3 | 4 | 5 |
| 1. | Emp\_Id | int | Довге ціле | Унікальний ідентифікатор співробітника |
| 2. | Emp\_Name | varchar | 0..20 | Ім'я співробітника |
| 3. | Emp\_Surname | varchar | 0..20 | Прізвище співробітника |
| 4. | Emp\_FatherName | varchar | 0..20 | По-батькові співробітника |
| 5. | Emp\_Position | varchar | 0..50 | Посада співробітника |
| 6. | Emp\_Salary | money |  | Зарплата співробітника |
| 7. | Emp\_JoiningDate | datetime |  | Дата прийняття співробітника на роботу |
| 8. | Emp\_Gender | char |  | Стать співробітника |

Таблиця 1.8 – Структура вхідних даних таблиці «Закупівлі»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № з/п | Ідентифікатор | Тип | Діапазон значень | Пояснення |
| 1 | 2 | 3 | 4 | 5 |
| 1. | Sti\_Id | int | Довге ціле | Унікальний ідентифікатор документу закупівлі |
| 2. | Sti\_Date | datetime |  | Дата надходження товару |
| 3. | Sti\_StockInBy | varchar | 0..50 | Ім'я особи, яка привезла товар |
| 4. | Sti\_Status | int | 1 | Статус надходження |
| 5. | Sti\_SupplierId | int | Довге ціле | Унікальний ідентифікатор постачальника |
| 6. | Sti\_IsConfirmed | bit | 0-1 | Підтвердження надходження товару |

Таблиця 1.9 – Структура вхідних даних таблиці «Постачальник»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № з/п | Ідентифікатор | Тип | Діапазон значень | Пояснення |
| 1 | 2 | 3 | 4 | 5 |
| 1. | Sup\_Id | int | Довге ціле | Унікальний ідентифікатор постачальника |
| 2. | Sup\_Name | varchar | 0..50 | Назва постачальника |
| 3. | Sup\_Address | text |  | Адреса постачальника |
| 4. | Sup\_ContactPerson | varchar | 0..50 | Контактна особа постачальника |
| 5. | Sup\_Phone | varchar | 0..50 | Телефон постачальника |
| 6. | Sup\_Email | varchar | 0..50 | Електронна адреса постачальника |
| 7. | Sup\_Fax | varchar | 0..50 | Факс постачальника |

Таблиця 1.10 – Структура вхідних даних таблиці «Користувач»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № з/п | Ідентифікатор | Тип | Діапазон значень | Пояснення |
| 1 | 2 | 3 | 4 | 5 |
| 1. | Usr\_Id | int | Довге ціле | Унікальний ідентифікатор користувача |
| 2. | Usr\_Employee | int | Довге ціле | Ідентифікатор співробітника, який є користувачем |
| 3. | Usr\_Username | varchar | 0..50 | Ім'я користувача |
| 4. | Usr\_Password | varchar | 0..50 | Пароль користувача |

Продовження таблиці 1.10

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| 5. | Usr\_Role | int | Довге ціле | Роль користувача (наприклад, адміністратор) |
| 6. | Usr\_IsActivate | bit | 0-1 | Чи активний обліковий запис користувача |

## Визначення структури вихідних даних

Вихідною інформацією задачі для користувача будуть звіти та форми, які будуть відображатися на під час роботи з програмою як таблиці та форми. Також вихідна інформація може бути роздрукована . Вихідні дані – це дані, які виконавець видає як результат здійсненої роботи. Вихідними даними у програмному додатку є інформація про товари, чеки, клієнтів та працівників. В таблиці 1.11 показані вихідні дані, які використовуються в дипломному проєкті.

Таблиця 1.11 - Вихідна інформація

|  |  |  |  |
| --- | --- | --- | --- |
| № | Назва | Форма | Звіт |
| 1 | Товари | + |  |
| 2 | Чек |  | + |
| 3 | Залишки на складі | + | + |
| 4 | Звіт Топ продаж | + | + |
| 5 | Транзакції покупок | + | + |
| 6 | Працівники | + | + |
| 7 | Звіт Прибуток | + | + |

Таблиця 1.12 - Структура звіту « Прибуток»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| №  з/п | Ідентифікатор | Тип | Діапазон  значень | Пояснення |
| 1. | Місяць | Текстовий | 0…50 | - |
| 2. | Сума | Грошовий | - | - |

Таблиця 1.13 - Структура звіту « Топ продаж»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| №  з/п | Ідентифікатор | Тип | Діапазон  значень | Пояснення |
| 1. | Код товару | Текстовий | 0…50 | - |
| 2. | Назва товару | Грошовий | - | - |
| 3. | Кількість | Числовий | Довге ціле | - |

Таблиця 1.14 - Структура чеку

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| №  з/п | Ідентифікатор | Тип | Діапазон  значень | Пояснення |
| 1. | Назва товару | Текстовий | 0…50 | - |
| 2. | Ціна | Грошовий | - | - |
| 3. | Кількість | Числовий | Довге ціле | - |
| 4. | Знижка % | Числовий | 0-100 | - |
| 5. | Знижка | Грошовий | - | - |
| 6. | Сума | Грошовий | - | - |
| 7. | Касир | Текстовий | 0-50 | - |
| 8. | Дата | Дата/час | - | - |
| 9. | Назва оптової бази | Текстовий | 0-50 | - |
| 10. | Адреса | Текстовий | 0-50 | - |