Міністерство освіти і науки України

Івано-Франківський національний технічний університет нафти і газу

Інститут інформаційних технологій

КУРСОВА РОБОТА

з дисципліни "Об'єктно-орієнтоване програмування"

на тему: Розробка проекту консольної бази даних по предметній області засобами С++

Студента 1 курсу групи ІП-22-1

напряму підготовки 121

Інженерія програмного забезпечення

Чурчак В.В.

Керівник: Доцент Шекета В.І.

Національна шкала \_\_\_\_\_\_\_\_\_\_\_\_\_

Кількість балів:\_\_\_\_\_ Оцінка: ECTS\_\_\_\_\_

Члени комісії

\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище та ініціали)

2023

Івано-Франківськ

**Івано-Франківський національний технічний університет нафти і газу Кафедра** *Інженерії програмного забезпечення*

**Дисципліна** *Об’єктно-орієнтоване програмування*

**Спеціальність** *Інженерія програмного забезпечення*

**Курс** 1 **Група** ІП-22-1 **Семестр** 2

**ЗАВДАННЯ**

**НА КУРСОВИЙ ПРОЕКТ (РОБОТУ)**

**Студенту** Чурчаку Володимиру Володимировичу

**(прізвище, ім’я, по батькові)**

***1.* Тема проекту (роботи):** *Розробка проекту консольної бази даних по предметній області засобами* ***С++***

**2. Термін здачі студентом закінченого проекту (роботи):** 1 червня 2023 р.

***3.* Вхідні дані до проекту (роботи) :** *Згідно з заданим варіантом*

***4.* Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити):** *Вступ, аналіз предметної області, постановка завдання, створення діаграми класів для розв’язування поставленої задачі, Бізнес логіка , висновок, список використаних джерел, додатки.*

|  |  |  |
| --- | --- | --- |
| Номер і назва етапів курсового проекту (роботи) | Термін виконання етапів проєкту (роботи) | Примітка |
| Вибір варіанту завдання. Теоретична довідка по предметній області | 28-30 квітня 2023р. | Виконано |
|  |  |  |
| Побудова рішення для виконання проекту. Обрання  методу програмування | 01-27 травня 2023р. | Виконано |
|  |  |  |
| Розробка та документування  проекту | 28-31 травня 2023р. | Виконано |
|  |  |  |
| Оформлення пояснювальної записки та компіляція кінцевого  проекту | . |  |
|  | . |  |
| Захист курсового проекту |  | Виконано |
|  |  |  |

КАЛЕНДАРНИЙ ПЛАН

**Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**(особистий підпис) (розшифровка підпису)**

**Керівник \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**(особистий підпис) (розшифровка підпису)**

**«\_\_\_» \_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_ р**

ЗМІСТ

ВСТУП………………………………………………………………………...2

1. ТЕОРЕТИЧНА ЧАСТИНА

1.1 Основні принципи ООП………………………………………………..3

1.2 Актуальність теми………………………………………………………4

2. ТЕХНІЧНА ЧАСТИНА

2.1 Постановка задачі……………………………………………………….5

2.2 Опис класів ……………………………………………………………...6

2.3 Структура меню…………………………………………………………7

3. КОД ПРОГРАМИ МОВОЮ С++………………………………………….8

ВИСНОВОК…………………………………………………………………...28

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ…………………………..…………29

ДОДАТОК A

ВСТУП

Програмування на С++ займає особливе місце серед сучасних мов програмування і широко використовується в розробці програмного забезпечення. Ця мова надає потужні інструменти для реалізації складних проектів, включаючи створення баз даних.

Метою даного курсового завдання є розгляд процесу створення консольної бази даних мовою програмування С++. Консольні бази даних є одним із основних типів баз даних, що надає можливість зберігати, оновлювати та отримувати дані за допомогою командного рядка. Це важливий елемент багатьох програм, особливо якщо дані потрібно обробляти без графічного інтерфейсу.

Крім того, у даній роботі пропонується розглянути концепцію об’єктно-орієнтованого програмування (ООП) та її використання в контексті створення баз даних. ООП вважається одним із найефективніших методів розробки програмного забезпечення, оскільки він дозволяє створювати модульний, гнучкий і багаторазово використовуваний код.

У даній курсовій роботі будуть розглянуті основні аспекти створення консольної бази даних з використанням мови програмування C++. Вона включатиме пояснення коду програми, опис класів та їх взаємодії, а також приклади використання консольної бази даних. Ми також розглянемо теоретичні відомості про об'єктно-орієнтоване програмування та його застосування в контексті створення баз даних.

Теоретична частина

1.1 Основні принципи ООП

Основні принципи об’єктно-орієнтованого програмування (ООП) — це основні концепції розробки програмного забезпечення з використанням об’єктно-орієнтованих методів. Вони надають можливість створювати модульний, гнучкий і багаторазово використовуваний код. Основні принципи ООП включають наступне:

1. Інкапсуляція (Encapsulation): Інкапсуляція в ООП відноситься до процесу упаковки даних і пов’язаних з ними методів в один об’єкт (клас). Інформація про дані та методи, які їх використовують, зберігається разом і зовнішні компоненти не можуть отримати до неї прямий доступ. Натомість дані взаємодіють через загальнодоступні методи, які забезпечують контроль доступу та забезпечують узгодженість даних.

2. Спадкування: Спадкування дозволяє створювати нові класи на основі існуючих класів. Клас, який успадковує властивості та методи від іншого класу, називається похідним класом, а клас, який успадковує властивості, називається базовим класом або суперкласом. Спадкування дозволяє використовувати та змінювати існуючий код, спрощуючи розробку та забезпечуючи повторне використання коду.

3. Поліморфізм: Поліморфізм дозволяє об’єктам різних класів виконувати ту саму операцію з різними результатами. Це означає, що ви можете мати багато класів, які реалізують однаковий інтерфейс (набір методів), але виконують різні дії. Поліморфізм спрощує роботу з об’єктами різних типів, дозволяючи обробляти їх так, ніби вони одного виду.

4. Абстракція: Абстракція в ООП відноситься до процесу вилучення важливих характеристик об’єкта, ігноруючи деталі, які не потрібні для виконання певних завдань. Це дозволяє визначити загальну об’єктну модель, яку можна використовувати для створення конкретних об’єктів. Абстракції допомагають зосередитися на основних аспектах проблеми та полегшують розробку та розуміння коду. Ці основні принципи ООП дозволяють розробникам писати більш структурований, читабельний і ефективний код. Застосування цих принципів до створення консольної бази даних допоможе покращити організацію коду, полегшити розширення та придатність коду в майбутньому.

1.1 Актуальність теми

Актуальність теми "Створення консольної бази даних з використанням мови програмування C++" полягає в її важливості та широкому застосуванні у сучасному програмуванні та розробці програмного забезпечення. Ось кілька основних причин, чому ця тема є актуальною:

1. Бази даних є необхідною складовою багатьох програмних проектів: Багато програм, незалежно від їхньої природи або розміру, вимагають збереження, оновлення та маніпулювання великими обсягами даних. Бази даних забезпечують структуроване зберігання даних і надають ефективні інструменти для доступу до них. Розуміння процесу створення бази даних з використанням мови C++ є важливим для розробки програм, які працюють зі значним обсягом даних.

2. Консольна база даних є важливою альтернативою графічному інтерфейсу: В деяких випадках, особливо у сфері системного адміністрування або веб-розробки, консольний інтерфейс є більш зручним та ефективним для взаємодії з базою даних. Створення консольної бази даних дозволяє зосередитися на самому функціоналі та оптимізації роботи з даними, ігноруючи складність графічного інтерфейсу.

3. Розуміння об'єктно-орієнтованого програмування (ООП) є важливим навичками для сучасних розробників: Мова програмування C++ використовує ООП підхід, який дозволяє створювати модульний і гнучкий код. ООП дозволяє зберігати код організованим, полегшує розширення та підтримку програми. Вивчення створення консольної бази даних з використанням C++ допоможе вам покращити навички ООП та зрозуміти принципи, які лежать в основі ефективної розробки програмного забезпечення.

4. Налагодження та оптимізація баз даних є важливими задачами: Розробка та оптимізація баз даних є процесом, який вимагає глибокого розуміння внутрішньої структури та принципів роботи бази даних. Створення консольної бази даних з використанням C++ надає можливість детально вивчити процеси збереження, оновлення та оптимізації даних. Це допоможе вам стати більш компетентним в роботі з базами даних та забезпечити їх ефективну роботу.

Технічна частина

2.1 Постановка задачі

В даній курсовій роботі необхідно розробити консольну базу даних для торгової організації, яка займається торгівлею в різних типах торгових точок

Торгові точки розміщують замовлення, які базуються на заявках, що надходять з кожної точки. Менеджери організації торгівлі вибирають постачальників, формують замовлення, в яких вказують найменування товарів та замовляють їх кількість, яка може відрізнятися від запиту торгової точки. Якщо назва товару раніше не була поставлена, вона додається до довідника номенклатури товарів. Ринок постачальників постійно вивчається, що може призводити до появи нових постачальників і зникнення старих. Торгова організація може отримувати одні й ті ж товари від різних постачальників за різними цінами.

Отримані товари розподіляються по торгових точках, і цей розподіл доступний в будь-який момент. Продавці в торгових точках ведуть облік продажу товарів, включаючи номенклатуру та кількість проданого товару. Універмаги та магазини також фіксують імена і характеристики покупців, дозволяючи вести облік покупців і їхніх покупок. Торгові точки мають можливість змінювати ціни на товари в залежності від попиту та пропозиції, а також передавати товари до інших торгових точок за погодженням.

Метою проекту є розробка консольної бази даних, яка забезпечуватиме ефективне управління торговими точками, замовленнями, постачальниками та обліком продажів. Система повинна забезпечувати можливість додавання, редагування та видалення інформації про торгові точки, товари, постачальників, замовлення і продажі. Крім того, вона повинна надавати зручний інтерфейс для перегляду та аналізу даних, включаючи звіти про продажі, облік товарів і замовлення постачальникам.

Розроблена база даних має сприяти покращенню ефективності і точності управління торговою організацією, спрощенню процесу замовлення і постачання товарів, а також забезпечувати зручний доступ до необхідної інформації для прийняття рішень.

2.1 Опис класів

Для ліпшої взаємодії з даними були створені класи. Клас «Product» використовується для даних про товар. Його властивості «.name» та «.allPointQuant» позначають назву товару та його загальну кількість постачання за увесь час. Дані для нього беруться з масиву «prod[]» та «quantPerProd[]».

Клас «Supplier» позначає постачальників товару. Дані для властивості «.name» він отримує з масиву «spec[]».

Для позначення покупця використовується клас «Customer». В ньому описані дані про найактивнішого покупця на кожній торгівельній точці. А саме вік, прізвище, ім’я, і кількість придбаного товару. Дані він отримує з масивів «customerName[]», «customerAge[]» та «buyingPerCustomer[]».

Найбільшим і найважливішим класом є «TradePoint», який позначає торгівельну точку і дані про неї. А саме: назву, кількість продуктів, яку постачали на точку, кількість продаж з цих продуктів, середню заробітню плату одного продавця, кількість торгових залів, рентабельність, відношення обсягів продаж до обсягів всього товару та орендну плату. Дані знаходяться в масивах quantPerPoint[], allSalesPerPoint[20], salesPerPoint[][], quantAllPerPoint[][], rent[] salarys[].

Для зручності масиви з загальними кількостями товарів були створені на основі двовимірних масивів з кількостями товарів по місяцях та торгівельних точках.

2.3 Структура меню

В даній роботі було використано оператори switch та case для реалізації вибору операцій, які повинна виконувати програма. При цьому, для перевірки даних на правильність вводу було застосовано оператори try та catch. Також з їхньою допомогою вдалося уникнути помилок, які пов’язані з безкінечною генерацією одного запиту.

При вводі неправильного номеру операції передбачено попередження користувача про його помилку і надання можливості виконати операцію ще раз. Ця функція доступна на всіх етапах і розгалуженнях запитів. Користувач також в будь-який момент може вийти в головне меню, після чого завершити роботу програми.

Для зручності інтерфейсу було використано табуляцію. Завдяки цьому користувачу буде зручніше розуміти, на якому етапі розгалуження зараз програма.

Спершу в програмі під’єднюються всі необхідні для коректної роботи бібліотеки. Далі оголошуються вищезгадані класи а за ними масиви і змінні choice, choice1, choice2, choice3, choice4 та input, які будуть використовуватися як вибір для операцій. Після цього починається основний цикл програми. Він буде діяти поки користувач не захоче вийти з програми. В даному циклі користувач обирає ті операції та вводить ті дані, які йому потрібні для отримання інформації з бази даних.

Для зручності подальшого редагування в коді наведені коментарі до кожного масиву даних.

В додатку A будуть наведені приклади виконання програми.

Код програми мовою C++

#include <iostream>

#include <vector>

#include <string>

#include <Windows.h>

#include <cmath>

#include <cstdlib>

#include <ctime>

// Оголошення класів

// Клас для опису товару

class Product {

public:

std::string name;

int allPointQuant;

};

// Клас для опису постачальника

class Supplier {

public:

std::string name;

};

// Клас для опису покупця

class Customer {

public:

std::string name;

int age;

int buyQuantity;

};

// Клас для опису торгової точки

class TradePoint {

public:

std::string name;

double productQuant;

int productSales;

int Salary;

int areaQuant;

double circ;

double rentable;

int rentPay;

};

// Головна функція програми

int main() {

//Додавання української мови//

SetConsoleOutputCP(1251);

SetConsoleCP(1251);

//Масиви даних//

//Назви постачальників

std::string spec[10] = { "ABC Supply Co.", "XYZ Corporation","Acme Suppliers" , "Global Trading Ltd","Mega Distributors", "Superior Products Inc.",

"Prime Merchandise", "Direct Sourcing Co.", "Quality Goods Ltd.", "Reliable Importers" };

//назви товарів

std::string prod[10] = { "Відеокарти","Смартфони","Павербанки","Ноутбуки","Навушники","Клавіатури","Комп`ютерні миші",

"Монітори","Системні блоки","Процесори" };

//назви торгових точок

std::string point[20] = { "Техносвіт","Електромагазин","Гаджетплаза","Смартпункт","Мобілайф","СмартГурт","Комп'ютерний раєтон",

"Технокомплект","Технолайф","ГаджетМастер","ЕлектроКуток","ПКХаб","СмартМаг","ТехноТрейд","Периферійна станція","Цифровий Всесвіт",

"ГаджетБазар","Електрофантастіка","ПКМагія","ЕлектроПК" };

//імена найактивніших покупців

std::string customerName[20] = {"Максимов Максим","Іванова Ірина","Сидоренко Сергій","Петренко Оксана","Коваленко Костянтин","Василенко Вікторія"

"Лисенко Людмила","Григоренко Григорій","Ткаченко Тетяна","Олійник Олександр","Кузьменко Катерина","Литвиненко Леонід","Шевченко Ірина","Кравченко Кирило","Романенко Роман",

"Мельник Марія","Павленко Павло","Гриценко Ганна","Савченко Сергій","Бондаренко Богдана"};

//вік найактвніших покупців

int customerAge[20] = { 28,32, 45, 36, 41, 22, 49, 29, 38, 47, 19, 25, 44, 31, 20, 33, 42, 23, 39, 30};

//кількість куплених товарів найактивнішими покупцями

int buyingPerCustomer[20]{

10023,25639,2789,1979,1783,3895,1785,2594,9452,4763,6592,1915,1747,1593,2911,3481,7582,8261,1125,8946

};

int salarys[20] = { 7254, 12456, 8934, 10123, 13456, 7845, 10987, 6321, 14234, 9765, 11876, 5134, 13789, 9654, 6412, 12345, 7465, 12987, 5768, 14678 };

//кількість торгових залів

int area[20] = { 3, 6, 4, 8, 5, 7, 2, 9, 4, 6, 8, 3, 7, 5, 9, 2, 10, 4, 6, 8 };

//Орендна плата

int rent[20] = { 71743, 34780, 46156, 38254, 46329, 40873, 39427, 37492, 47981, 45921, 36091, 42034, 44019, 49235, 39347, 41354, 33892, 40722, 37050, 35283 };

//Назви міяців

std::string month[12] = { "Січень", "Лютий", "Березень", "Квітень", "Травень", "Червень", "Липень", "Серпень", "Вересень", "Жовтень", "Листопад", "Грудень" };

//простачання товарів кожного постачальника протягом міяця загалом

int quantAll[10][12] = {

{29167, 31250, 27083, 30000, 25000, 28333, 29167, 30833, 26250, 28750, 32917, 27500},

{60000, 64167, 56667, 60833, 58333, 62500, 63333, 55833, 57500, 59167, 62500, 54167},

{46667, 51667, 48333, 45833, 50000, 52500, 47500, 49167, 47500, 51667, 49167, 53333},

{81667, 73333, 82500, 84167, 80000, 79167, 85833, 78333, 80000, 86667, 75833, 81667},

{20833, 21500, 18750, 20000, 17500, 19167, 20417, 18750, 19583, 22083, 18750, 20000},

{70000, 73333, 68333, 72500, 70833, 73333, 70833, 69167, 75000, 70833, 72500, 66667},

{39167, 40000, 36667, 39167, 41250, 38333, 40833, 37500, 40000, 41250, 38333, 40000},

{57500, 57917, 57917, 55000, 57500, 60833, 56250, 57500, 59167, 57500, 57917, 59167},

{43333, 43333, 46667, 43333, 46667, 46667, 43333, 46667, 43333, 46667, 43333, 46667},

{74167, 74167, 77500, 75000, 79167, 75000, 76667, 78333, 74167, 78333, 75000, 75000}

};

//простачання товарів кожного постачальника протягом міяця на кожній торгівельній точці

int quantAllPerPoint[20][12] = {

{14583, 15625, 13541, 15000, 12500, 14166, 14583, 15416, 13125, 14375, 16458, 13750},

{14584, 15625, 13542, 15000, 12500, 14167, 14584, 15417, 13125, 14375, 16459, 13750},

{30000, 32083, 28333, 30416, 29166, 31250, 31666, 27916, 28750, 29583, 31250, 27083},

{30000, 32084, 28334, 30417, 29167, 31250, 31667, 27917, 28750, 29584, 31250, 27084},

{23333, 25833, 24166, 22916, 25000, 26250, 23750, 24583, 23750, 25833, 24583, 26666},

{23334, 25834, 24167, 22917, 25000, 26250, 23750, 24583, 23750, 25834, 24584, 26667},

{40833, 36666, 41250, 42083, 40000, 39583, 42916, 39166, 40000, 43333, 37916, 40833},

{40834, 36667, 41250, 42084, 40000, 39584, 42917, 39167, 40000, 43334, 37917, 40834},

{10416, 10750, 9375, 10000, 8750, 9583, 10208, 9375, 9791, 11041, 9375, 10000},

{10417, 10750, 9375, 10000, 8750, 9584, 10209, 9375, 9792, 11042, 9375, 10000},

{35000, 36666, 34166, 36250, 35416, 36666, 35416, 34583, 37500, 35416, 36250, 33333},

{35000, 36667, 34167, 36250, 35417, 36667, 35417, 34584, 37500, 35417, 36250, 33334},

{19583, 20000, 18333, 19583, 20625, 19166, 20416, 18750, 20000, 20625, 19166, 20000},

{19584, 20000, 18334, 19584, 20625, 19167, 20417, 18750, 20000, 20625, 19167, 20000},

{28750, 28958, 28958, 27500, 28750, 30416, 28125, 28750, 29583, 28750, 28958, 29583},

{28750, 28959, 28959, 27500, 28750, 30417, 28125, 28750, 29584, 28750, 28959, 29584},

{21666, 21666, 23333, 21666, 23333, 23333, 21666, 23333, 21666, 23333, 21666, 23333},

{21667, 21667, 23334, 21667, 23334, 23334, 21667, 23334, 21667, 23334, 21667, 23334},

{37083, 37083, 38750, 37500, 39583, 37500, 38333, 39166, 37083, 39166, 37500, 37500},

{37084, 37084, 38750, 37500, 39584, 37500, 38334, 39167, 37084, 39167, 37500, 37500}

};

//продажі товарів щомісяця на кожній торговій точці

int salesPerPoint[20][12]{

{10897, 15000, 9950, 12498, 10834, 11849, 13894, 9236, 12453, 12478, 12365, 10986},

{11876, 12987, 9486, 14856, 12359, 13589, 10184, 12548, 12085, 13058, 13047, 13700},

{28056, 26887, 26499, 15879, 26498, 22198, 25496, 24895, 27581, 14835, 30894, 26891},

{28989, 28649, 28159, 29888, 26879, 21215, 18989, 26897, 27853, 29459, 30251, 26414},

{20333, 24558, 23858, 20844, 20363, 23468, 18946, 15643, 20849, 18899, 20684, 26558},

{21705, 25613, 23541, 22644, 24998, 21847, 22849, 24156, 20849, 18548, 10949, 25879},

{40598, 16846, 39478, 38494, 38940, 35412, 33589, 12589, 14849, 22560, 27846, 30516},

{31549, 35941, 40589, 25879, 19135, 36948, 40849, 35479, 32094, 42489, 36894, 32944},

{9153, 1785, 7816, 2594, 5674, 1799, 10000, 1798, 7895, 10548, 5492, 6985},

{1954, 9486, 2587, 8885, 3561, 8541, 10001, 7484, 9584, 6284, 3581, 1254},

{26588, 34589, 25942, 35985, 24916, 28941, 19815, 31549, 35894, 32654, 22189, 29415},

{32894, 36589, 31158, 34856, 31894, 33359, 32551, 33891, 24855, 26941, 35418, 23186},

{15987, 16654, 15998, 18915, 20589, 19000, 15984, 12549, 18000, 17805, 15810, 14785},

{15798, 12658, 17885, 19350, 20023, 17503, 20200, 17512, 15963, 20489, 13549, 18533},

{25943, 28469, 27486, 15865, 28456, 12348, 15266, 15556, 29958, 28456, 21536, 22225},

{27563, 21786, 27484, 22598, 22549, 12865, 28120, 4156, 19156, 28451, 11536, 7485},

{20000, 20646, 5161, 20498, 23000, 5198, 20599, 18654, 21359, 18798, 15843, 19874},

{20796, 13598, 14862, 12849, 22684, 23158, 20589, 15788, 19488, 17563, 21445, 23331},

{35846, 34126, 30568, 37000, 35618, 31248, 35919, 32126, 25846, 38461, 15649, 24953},

{4986, 12169, 34599, 34118, 8469, 28448, 24594, 28496, 33892, 39150, 18880, 28946}

};

//Створення масивів на основі попередніх масивів

//загальна кількість товару кожного постачальника

int quantPerProd[10];

int j = 0;

for (int i = 0; i <= 9; i++) {

quantPerProd[i] = 0;

for (int k = 0; k <= 11;k++) {

quantPerProd[i] = quantPerProd[i] + quantAll[j][k];

};

j++;

}

//загальна кількість товару на кожній торгівельній точці

int quantPerPoint[20];

j = 0;

for (int i = 0; i <= 19; i++) {

quantPerPoint[i] = 0;

for (int k = 0; k <= 11; k++) {

quantPerPoint[i] = quantPerPoint[i] + quantAllPerPoint[j][k];

}

j++;

}

//кількість проданого товару на кожній торгівельній точці

int allSalesPerPoint[20];

j = 0;

for (int i = 0; i <= 19; i++) {

allSalesPerPoint[i] = 0;

for (int k = 0; k <= 11; k++) {

allSalesPerPoint[i] = allSalesPerPoint[i] + salesPerPoint[j][k];

}

j++;

}

//головний цикл програми

int choice,choice1,choice2,choice3,choice4;

do {

std::cout << "Меню:" << std::endl;

std::cout << "1. Отримати дані про товар" << std::endl;

std::cout << "2. Отримати дані про торгівельну точку" << std::endl;

std::cout << "3. Вийти з програми" << std::endl;

std::cout << std::endl;

std::cout << "Виберіть опцію: ";

std::string input;

std::cin >> input;

choice = std::stoi(input);

if (choice >= 1 && choice <= 3) {

switch (choice) {

case 1: {

std::cout << "\tМеню:" << std::endl;

for (int i = 0; i <= 9; i++) {

std::cout <<"\t" << i << ". " << prod[i] << std::endl;

}

std::cout << std::endl;

std::cout << "\tВиберіть назву товару за номером або введіть число 11, щоб повернутися назад ";

TradePoint selectFirstPoint, selectSecondPoint;

std::cin >> input;

choice = std::stoi(input);

if (choice >= 0 && choice <= 9) {

Product selectProduct;

selectProduct.name = prod[choice];

std::cout<<"\t\t" << selectProduct.name << std::endl;

std::cout << "\t\tМеню:" << std::endl;

std::cout << "\t\t1. Отримати загальну кількість товару, яку постачали за увесь період" << std::endl;

std::cout << "\t\t2. Отримати загальну кількість товару, яку постачали на певну точку" << std::endl;

std::cout << "\t\t3. Отримати назву постачальника" << std::endl;

std::cout << "\t\t11. Вийти з програми" << std::endl;

std::cout << std::endl;

std::cout << "\t\tВиберіть опцію: ";

std::cin >> input;

choice1 = std::stoi(input);

if (choice1 >= 1 && choice1 <= 3) {

switch (choice1) {

case 1:{

selectProduct.allPointQuant=quantPerProd[choice];

std::cout << "\t\tЗагальна кількість товару "<<selectProduct.name<<" на всіх торгових точках дорівнює " << selectProduct.allPointQuant<<" одиниць" << std::endl;

break;

}

case 2: {

selectFirstPoint.name = point[choice\*2];

selectSecondPoint.name = point[(choice\*2)+1];

std::cout << "\t\tСписок торгових точок" << std::endl;

std::cout <<"\t\t1. "<< selectFirstPoint.name << std::endl;

std::cout << "\t\t2. "<<selectSecondPoint.name << std::endl;

std::cout << "\t\tВиберіть торгову точку або введіть число 11, щоб повернутися назад";

std::cin >> input;

choice1 = std::stoi(input);

if (choice1 >= 1 && choice1 <= 2) {

switch (choice1) {

case 1: {

selectFirstPoint.productQuant=quantPerPoint[choice\*2];

std::cout<<"\t\tЗагальна кількість товару " << selectFirstPoint.productQuant<<" на торговій точці "<< selectFirstPoint.name << std::endl;

break;

}

case 2: {

selectSecondPoint.productQuant = quantPerPoint[(choice \* 2) + 1];

std::cout << "\t\tЗагальна кількість товару " << selectSecondPoint.productQuant << " на торговій точці " << selectSecondPoint.name << std::endl;

break;

}

}

}

else if (choice1 == 11) {

break;

}

else {

std::cout << "\t\tНекоректний вибір. Спробуйте ще раз." << std::endl;

break;

}

}

case 3: {

Supplier selectedSupplier;

selectedSupplier.name = spec[choice];

std::cout << "\t\tНазва постачальника товару "<<selectProduct.name<<" - "<<selectedSupplier.name << std::endl;

break;

}

}

break;

}

else if (choice1 == 11) {

break;

}

else {

std::cout << "\t\tНекоректний вибір. Спробуйте ще раз." << std::endl;

break;

}

}

else if (choice==11) {

break;

}

else {

std::cout << "\t\tНекоректний вибір. Спробуйте ще раз." << std::endl;

}

}

case 2: {

TradePoint selectedPoint;

std::cout << "\tМеню:" << std::endl;

for (int i = 0; i <= 19; i++) {

std::cout <<"\t"<< i << ". " << point[i] << std::endl;

}

std::cout << std::endl;

std::cout << "\tВиберіть назву торгівельної точки за номером або введіть число 20, щоб повернутися назад ";

std::cin >> input;

choice2 = std::stoi(input);

if (choice2 >= 0 && choice2 <= 19) {

selectedPoint.name = point[choice2];

std::cout <<"\t\t"<< selectedPoint.name << std::endl;

std::cout << "\t\tМеню:" << std::endl;

std::cout << "\t\t1. Отримати загальну кількість товару, яку продали за увеь час" << std::endl;

std::cout << "\t\t2. Отримати загальну кількість товару, яку продали за певний місяць"<< std::endl;

std::cout << "\t\t3. Отримати загальні дані про найактивнішого користувача" << std::endl;

std::cout << "\t\t4. Отримати дані про заробітню плату одного продавця" << std::endl;

std::cout << "\t\t5. Отримати дані про рентабельність" << std::endl;

std::cout << "\t\t6. Отримати дані про товарообіг" << std::endl;

std::cout << std::endl;

std::cout << "\t\tВведіть номер операції або число 11, щоб повернутися назад " ;

std::cin >> input;

choice3 = std::stoi(input);

if (choice3 >= 1 && choice3 <= 6) {

switch (choice3) {

case 1: {

selectedPoint.productSales = allSalesPerPoint[choice2];

std::cout <<"\t\tЗагальна кількість товару, яку продали за увеь час на торговій точці "<< selectedPoint.name<<" дорівює " << selectedPoint.productSales << std::endl;

break;

}

case 2: {

std::cout << "\t\tМеню:" << std::endl;

for (int i = 0; i <= 11; i++) {

std::cout <<i<<". " << month[i] << std::endl;

}

std::cout << std::endl;

std::cout << "\t\tВведіть номер місяця або число 12, щоб повернутися назад ";

std::cin >> input;

choice4 = std::stoi(input);

if (choice4 >= 0 && choice4 <= 11) {

selectedPoint.productSales = salesPerPoint[choice3][choice4];

std::cout << "\t\tЗагальна кількість товару, яку продали в місяці " << month[choice4] << " дорівює " << selectedPoint.productSales << std::endl;

break;

}

else if (choice4 == 12) {

break;

}

else {

std::cout << "\t\tНекоректний вибір. Спробуйте ще раз." << std::endl;

}

}

case 3: {

Customer selectCustomer;

selectCustomer.name = customerName[choice2];

selectCustomer.age = customerAge[choice2];

selectCustomer.buyQuantity = buyingPerCustomer[choice2];

std::cout << "\tНайактивніший покупець" << std::endl;

std::cout << "\tПрізвище та ім' - " << selectCustomer.name << "" << std::endl;

std::cout << "\tВік - " << selectCustomer.age << "" << std::endl;

std::cout << "\tКількість придбаного товару - "<<selectCustomer.buyQuantity<<"" << std::endl;

break;

}

case 4: {

selectedPoint.Salary = salarys[choice2];

std::cout << "\t\tЗаробітня плата одного продавця дорівнює " << selectedPoint.Salary<< " грн" << std::endl;

break;

}

case 5: {

selectedPoint.productSales = allSalesPerPoint[choice2];

selectedPoint.rentPay = rent[choice2];

selectedPoint.Salary = salarys[choice2];

selectedPoint.areaQuant = area[choice2];

selectedPoint.rentable = selectedPoint.rentPay+(selectedPoint.areaQuant \* selectedPoint.Salary);

std::cout << "\t\tКількість витрат на продажу товару дорівнює " << selectedPoint.rentable<< " грн" << std::endl;

break;

}

case 6: {

selectedPoint.productSales = allSalesPerPoint[choice2];

selectedPoint.productQuant = quantPerPoint[choice2];

selectedPoint.circ = (selectedPoint.productSales/ selectedPoint.productQuant)\*100;

std::cout <<"\t\tВідношення проданого товару до його повної кількості дорівнює " << selectedPoint.circ<<"%" << std::endl;

break;

}

}

break;

}

else if (choice3 == 11) {

break;

}

else {

std::cout << "Некоректний вибір. Спробуйте ще раз." << std::endl;

}

}

else if (choice2 == 20) {

break;

}

else {

std::cout << "Некоректний вибір. Спробуйте ще раз." << std::endl;

}

}

case 3: {

std::cout << "Програма завершила роботу." << std::endl;

break;

}

}

}

else {

std::cout << "Некоректний вибір. Спробуйте ще раз." << std::endl;

}

std::cout << std::endl;

}while (choice != 3);

return 0;

}

Висновки

У ході виконання курсової роботи була розроблена консольна база даних для торгової організації, яка веде торгівлю в різних типах торгових точок.

Розроблена база даних дозволяє зберігати інформацію про різні характеристики торгових точок, замовленнях, товарах і постачальниках. Крім того, система забезпечує генерацію звітів і аналітичних даних для прийняття рішень.

Розроблена база даних виявляється актуальною і корисною для торгових організацій. Вона дозволяє покращити ефективність управління, спростити процес замовлення та постачання товарів, а також забезпечити зручний доступ до важливої інформації.

Протягом виконання курсової роботи були отримані практичні навички роботи з об'єктно-орієнтованою моделлю програмування та базами даних. Розробка цієї бази даних дала змогу глибше зрозуміти принципи ООП та їх застосування в реальних проєктах.

У підсумку, розроблена база даних є важливим інструментом для ефективного управління торговою організацією, спрощення процесу замовлення і постачання товарів, а також забезпечує зручний доступ до необхідної інформації для прийняття рішень. Вона може бути використана як основа для подальших розширень та вдосконалень у сфері управління торговою діяльністю.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Bjarne Stroustup Tour of C++:книга - Addison-Wesley Professional, 2022 – 320 с.

1. Stenley B. Lippman C++ Primer: книга - Addison-Wesley, 2012 - 976 с.
2. You Tube канал: https://www.youtube.com/@SimpleCodeIT

ДОДАТОК А

