Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Кафедра автоматизації проектування  
енергетичних процесів і систем

**КУРСОВА РОБОТА**

з дисципліни «Об’єктно-орієнтоване програмування - 1»

(назва дисципліни)

на тему:

«Створення моделей об’єктів реального світу на мові C++»

Студента групи **ТІ-81**

спеціальність **121 «Інженерія програмного забезпечення»**

освітня програма **Інженерія програмного забезпечення розподілених систем**

Соломахи О.О.

(прізвище та ініціали)

Керівник доцент, канд. фіз.-мат. наук Карпенко С.Г.

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Національна оцінка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кількість балів: \_\_\_\_\_\_\_\_\_\_Оцінка: ECTS \_\_\_\_\_

Члени комісії \_\_\_\_\_\_\_\_\_\_\_ доц., к. ф.-м. н. Карпенко С.Г.

(підпис) (вчене звання, науковий ступінь, прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_ доц., к.е.н. Левченко Л.О.

(підпис) (вчене звання, науковий ступінь, прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_ доц., к.т.н. Шаповалова С.І.

(підпис) (вчене звання, науковий ступінь, прізвище та ініціали

Київ - 2019 рік

**Національний технічний університет України**

**“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет ТЕПЛОЕНЕРГЕТИЧНИЙ ФАКУЛЬТЕТ

( повна назва)

Кафедра автоматизації проектування енергетичних процесів та систем

(повна назва)

рівень вищої освіти ***перший (бакалаврський)***

спеціальність ***121 «Інженерія програмного забезпечення»***

(шифр і назва)

освітня програма ***Інженерія програмного забезпечення розподілених систем***

(назва)

ЗАВДАННЯ

**НА КУРСОВУ РОБОТУ СТУДЕНТУ**

Соломасі Олександру Олександровичу

(прізвище, ім’я, по батькові)

1. Тема роботи «Створення моделей об’єктів реального світу на мові C++»

керівник курсової роботи –

Карпенко Станіслав Григорович, канд. фіз.-мат. наук, доцент

( прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

2. Строк подання студентом роботи – 24 грудня 2019 р.

3. Вихідні дані до проекту (роботи): мова C++, модель об’єкту реального світу.

4. Зміст пояснювальної записки курсової роботи (перелік питань, які потрібно розробити) – Розробити та обґрунтувати ієрархічну модель класів для опису об’єкту реального світу, використовуючи інструментальні можливості мови C++, зокрема успадковування, інкапсуляцію, перевантажені операції, шаблони функцій, запис у двійкові файли та обробку виключень.

5. Перелік графічного матеріалу –

графічне зображення ієрархічної моделі класів з урахуванням як успадковування, так й включення; відображення значень основних параметрів реальних об’єктів у вигляді таблиці.

6. Дата видачі завдання – 24 вересня 2019

**КАЛЕНДАРНИЙ ПЛАН**

|  |  |  |  |
| --- | --- | --- | --- |
| №  з/п | Назва етапів виконання курсової роботи | Строк виконання етапів роботи | Примітка |
|  | Побудова ієрархічної моделі класів | 15.10.2019 |  |
|  | Розробка та застосування інструментальних засобів обробки даних | 19.11.2019 |  |
|  | Написання розрахунково-пояснювальної записки | 24.12.2019 |  |

**Студент \_\_\_\_\_\_\_\_\_ Соломаха О.О.**

( підпис ) (прізвище та ініціали)

**Керівник курсової роботи \_\_\_\_\_\_\_\_\_ Карпенко С.Г.**

( підпис ) (прізвище та ініціали)

**АНОТАЦІЯ**

Було розроблено та описано ієрархічну модель класів для опису об’єктів реального світу, використовуючи інструментальні можливості мови C++, зокрема успадковування, інкапсуляцію, агрегацію, композицію перевантажені операції, шаблони функцій, запис у двійкові файли та обробку виключень.

Предметною областю завдання була частина інтернет магазину, зокрема об’єкти класу Покупця, Товару, Денного звіту, Замовлення.

Програма зроблена зручною для користувача, передбачено обробку виключних ситуацій та валідацію даних.

**ANNOTATION**

Has been developed and documented hierarchy model of classes for describing real world objects using C++ abilities such as inheritance, encapsulation, overloaded operators, aggregation, composition, template functions, write to binary files and exception handling.

Domain of the task was creating a part of online shop, including such classes as Customer, Product, Daily Report, Order.

Program was made handful for users, includes exception handling and input validation.

**Оформлення курсової роботи**

**ЗМІСТ**

1. ВСТУП 5

ОСНОВНА ЧАСТИНА 6

* + 1. 1. Взаємозв'язки між класами 6
    2. 2. Обмеження для полів класів 6
    3. 3. Перевантажені операції 6
    4. 4. Обробка виключних ситуацій 6
    5. 5. Шаблони функцій та їх параметри 6

1. ВИСНОВКИ 15

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 16

ДОДАТКИ 17

Додаток 1. Текст програми 7

* + 1. Додаток 2. Опис програми 7

Зм.

Лист

№ докум.

Підпис

Дата

Аркуш

1

УКР.НТУУ “КПІ імені І.Сікорського”\_ТЕФ\_АПЕПС\_TІ81147 3 КР 81-1

Розробив

Соломаха О.О.

Перевірив

Карпенко С.Г.

Рецензент.

Норм Контр.

Кузьміна Л.П.

Затвердив

Коваль О.В.

Назва теми

«Створення моделей об’єктів реального світу на мові C++»

Лит.

Аркушів

1

ТЕФ

Изм

Аркуш

№ докум

Подп

Дата

Арк

**ВСТУП**

Курсова робота – заключний і найважливіший етап вивчення і застосування дисципліни «Об’єктно-орієнтоване програмування - 2». Її виконання передбачає поглиблення навичок як застосування теоретичних знань, так і розв’язування поставленої задачі з проходженням усіх етапів від формалізації задачі і її корегування до реалізації на комп’ютері.

Виконання роботи передбачає знання таких теоретичних матеріалів:

* Клас - визначає абстрактні характеристики деякої сутності, включно з характеристиками самої сутності (її атрибутами або властивостями) та діями, які вона здатна виконувати (її поведінкою, методами або можливостями). Класи вносять модульність та структурованість в об'єктно-орієнтовану програму. Зазвичай клас має бути зрозумілим для не-програмістів, що знаються на предметній області, що, у свою чергу, значить, що клас повинен мати значення в контексті. Також, код реалізації класу має бути досить самодостатнім. Властивості та методи класу, разом називаються його членами.
* Об'єкт - окремий екземпляр класу (створюється після запуску програми і ініціалізації полів класу). Сукупність значень атрибутів окремого об'єкта називається станом.
* Метод - можливості об'єкта.
* Успадкування (наслідування) - Клас може мати «підкласи», спеціалізовані, розширені версії надкласу. Можуть навіть утворюватись цілі дерева успадкування.
* Приховування інформації (інкапсуляція) - Приховування деталей про роботу класів
* від об'єктів, що їх використовують або надсилають їм повідомлення. Інкапсуляція досягається шляхом вказування, які класи можуть звертатися до членів об'єкта. Як наслідок, кожен об'єкт надає кожному іншому класу певний інтерфейс — члени, доступні іншим класам. Інкапсуляція потрібна для того, аби запобігти використанню користувачами інтерфейсу тих частин реалізації, які, швидше за все, будуть змінюватись. Це дасть змогу полегшити внесення змін без потреби змінювати і користувачів інтерфейсу.

Изм

Аркуш

№ докум

Подп

Дата

Арк

* Абстрагування - Спрощення складної дійсності шляхом моделювання класів, що відповідають проблемі, та використання найприйнятнішого рівня деталізації окремих аспектів проблеми.
* Поліморфізм - Поліморфізм означає залежність поведінки від класу, в якому ця поведінка викликається, тобто, два або більше класів можуть реагувати по-різному на однакові повідомлення.

Основними задачами курсової роботи є:

* Розробка ієрархічної моделі досліджуваного об’єкту.
* Визначення конкретних властивостей полів класів.
* Визначення та розробка перевантажених операторів.
* Визначення можливих застосувань виключних ситуацій.
* Розробка методів зчитування та запису інформації на жорсткий диск.
* Визначення та розробка необхідних делегатів.
* Розробка інтерфейсу користувача та тестування програми.

**ОСНОВНА ЧАСТИНА**

Изм

Аркуш

№ докум

Подп

Дата

Арк

**1. Взаємозв’язки між класами**

UML-діаграма побудованої ієрархії класів наведена на рисунку 1.1. Для спрощення розуміння в діаграмі наведено лише основні члени класів.

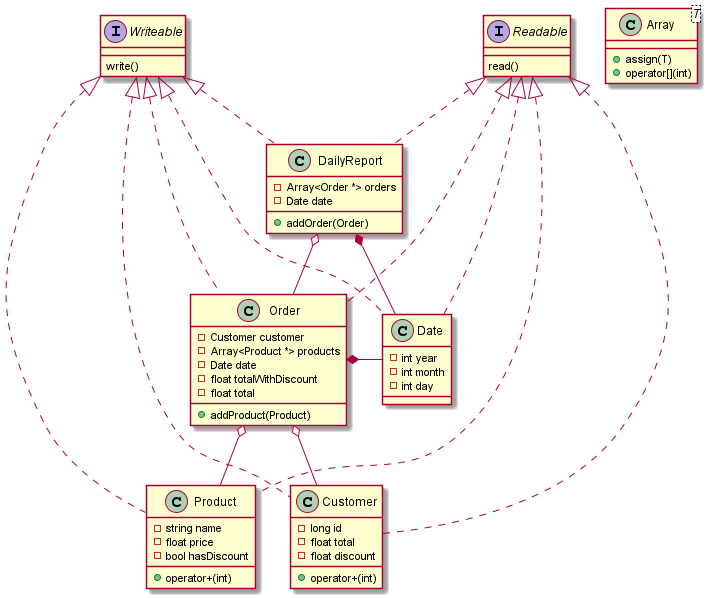


Рисунок 1.1.1. UML діаграма

Для більшої зрозумілості наведено таблицю з необхідними класами та їх членами, назвою українською та англійською.

Изм

Аркуш

№ докум

Подп

Дата

Арк

Таблиця 1.1.1. Таблиця структури класів

|  |  |  |  |
| --- | --- | --- | --- |
| Назва | Назва англійською | Члени | Назви членів англійською |
| Дата | Date | Рік, Місяць, День | Year, month, day |
| Постійний покупець | Customer | Ідентифікаційний номер, Загальна сума здійснених купівель, відсоток знижок | Id, total, discount |
| Товар | Product | Назва товару, Ціна товару, наявність знижки | Name, price, hasDiscount |
| Операція купівлі (Замовлення) | Order | Клієнт, Список товарів, Загальна сума товарів, Внесена сума з урахуванням знижки, Дата | Customer, products, total, date, totalWithDiscount |
| Денний звіт | Daily Report | Список операцій купівель, Дата | Orders, date |

З діаграми видно, що основні класи реалізують (наслідуються) від інтерфейсу (абстрактного класу) Writable та Readable, що мають віртуальні функції для запису та зчитування з файлу у двійковому форматі.

Для класу Дата було використано такий тип зв’язку як «композиція». З означення терміну слідує, що об’єкт класу Дата не може існувати без свого контейнеру (Операція купівлі, Денний звіт), тобто зі знищенням контейнеру буде і знищено об’єкт класу Дата.

Між об’єктами класу Замовлення та Товар і Покупець було використано тип зв’язку агрегація. З означення терміну «агрегація» слідує, що два об’єкти можуть існувати незалежно один від одного, тобто якщо контейнер (Замовлення) буде знищено, Товар та Покупець залишаться (наприклад, у базу даних). Такий самий тип зв’язку має і Денний звіт з Замовленням.

Усі поля класів є приватними. Для отримання та змінення полів класів реалізовано спеціальні методи – геттери та сеттери, у яких було реалізовано валідацію даних (див. пункт 2).

Изм

Аркуш

№ докум

Подп

Дата

Арк

Контейнером для динамічного масиву слугує об’єкт класу Array з реалізацією у шаблонному вигляді.

**2. Обмеження для полів класів**

Изм

Аркуш

№ докум

Подп

Дата

Арк

Відповідно до завдання необхідно було реалізувати обмеження для полів класів. Також, цей процес ще називається «валідацією даних».

Валідація даних необхідна для правильного зберігання та обробки інформації, цілісності системи, правильності введення даних.

Для обмежень значень полів класів було реалізовано методи, які перевіряють передані дані. Якщо введені дані не входять до заданих меж, виникає виключна ситуація, котру потрібно правильно обробити, вивести користувачеві повідомлення та діяти надалі, як того потребує завдання. Реалізовувати логіку валідації тільки у сеттерах не раціонально, бо не має можливості використати сеттери у «member initializer lists» у конструкторах.

Далі наведено використані обмеження значень полів класів:

* Рік – не раніше 1980 та не пізніше 2100
* Місяць – не менше 1 та не більше 12
* День – не менше 1 та не більше 31
* Ідентифікаційний номер – не менше за 0
* Загальна сума здійснених купівель - не менше за 0
* Відсоток знижок – десятковий дріб від 0 до 1 включно
* Назва товару – не більше 25 символів
* Ціна – не менше за 0
* Має знижку – 1 або 0 (так або ні)
* Загальна сума товарів – не менше за 0
* Внесена сума з урахуванням знижки – не менше за 0

**3. Перевантажені операції**

Изм

Аркуш

№ докум

Подп

Дата

Арк

Перевантажені операції дозволяють зручно використовувати вбудовані у мову програмування С++ операції, наприклад, операція додавання, вводу\виводу у потік, інкременту та декремнту, та багато іншого.

Для об’єктів класів, що мають динамічний масив, тобто є контейнерами, реалізовано оператори індексування. Класи з перевантаженою операцією

індексування – Операція купівлі, Денний звіт.

Оскільки тема курсової роботи має фінансовий характер, доцільним було перевантажити оператори додавання цілого числа та дробу. Для об’єкту класу Товар додавання цілого числа або десяткового дробу використовується для збільшення або зменшення вартості. Додавання до об’єкту класу Покупець збільшить його загальну суму здійснених купівель. Додавання до об’єкту класу Замовлення збільшить його поле «Внесена сума з урахуванням знижки».

Для основних класів було перевантажено оператор потокового виводу для виведення короткої інформації про об’єкт з основними полями.

Для основних було перевантажено оператор потокового вводу для отримання інформації з клавіатури користувача з обробкою виключних ситуацій.

Приклад перевантаженого оператору «+» для класу Товар:  
template<class T>  
Product Product::operator+(T plusPrice) {  
 price += plusPrice;  
 return \*this;  
}

**4. Обробка виключних ситуацій**

Изм

Аркуш

№ докум

Подп

Дата

Арк

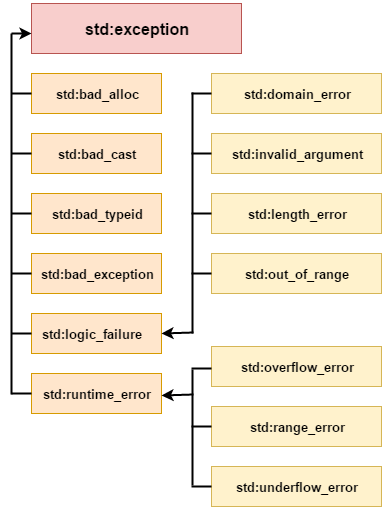
При роботі програми можуть виникати різні виключні ситуації. Наприклад, файл, що обрав користувач, не існує, неправильно введені дані, помилка виділення пам’яті, ділення на 0, тощо.

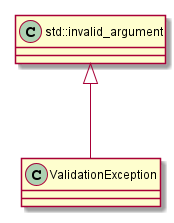
Для правильної поведінки програми в цих ситуаціях необхідно обробляти їх та повідомлювати користувача про подальші дії.

Використання виключень зменшує кількість коду по зрівнянню з повертанням коду з функції.

У цій роботі створення штучних виключних ситуацій було активно використано для валідації даних. Щоб відокремлювати виключення такого типу було створено окремий клас ValidationException, що наслідується від

std::invalid\_argument та передбачає установки текстового повідомлення користувачу. std::invalid\_argument наслідується від базового std::exception. Присутня обробка і виключень під час роботи з файлами.

 Під час обробки валідаційних виключень робота програми продовжується, користувачеві виводиться повідомлення про помилку та еще одна спроба ввести дані.



**5. Шаблони функцій та їх параметри**

Изм

Аркуш

№ докум

Подп

Дата

Арк

Шаблони функцій призначені для кодування узагальненого коду, не прив’язуючись до конкретного типу даних. У С++ можливе створення шаблонів функцій та шаблонів класу.

У цій роботі шаблони функцій та класів було використано у наступних місцях:

* при підрахунку середнього значення кількості товарів в операціях купівель – averageProductsCount();
* при підрахунку середнього значення загальної суми товарів в операціях за день – averageTotal();
* для об’єкту класу динамічного масиву, щоб була підтримка об’єктів різного типу (у роботі – для Замовлень і Товарів);
* перевантажений оператор «+» для Клієнта та Товару.

Наведені приклади дозволяють використовувати різні типи числових даних (int, float, double) для заданих функцій.

**ВИСНОВКИ**

Метою цієї роботи було практичне та теоретичне закріплення пройденого курсу «Об’єктно-орієнтоване програмування» використовуючи мову С++.

Изм

Аркуш

№ докум

Подп

Дата

Арк

Розроблено програму, що відповідає умовам завдання, зручна для користувача, вміє правильно оброблювати виключні ситуації.

Наведена у завданні структура класів є прикладом простої ієрархії, що потребує базові навички об’єктного стилю програмування. Також, запропонована тема «Інтернет магазин» відрізняється від прикладних комерційних аналогів, де в першу чергу акцент йде на структуру базу даних, у якій не завжди можливо і не завжди доцільно реалізувати базові аспекти об’єктно-орієнтованого програмування.

Введення даних через консоль не є зручним для користувача і потребує розвинутих навичок у цільової аудиторії в тому числі і для запуску програми.

**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

Изм

Аркуш

№ докум

Подп

Дата

Арк

1. 1. Г. Буч. Объектно – ориентированный анализ и проектирование с примерами приложений на С++, 2-е изд./ Пер. с англ. – М.: "Издательство Бином". 1998 г. – 560 с., ил.
2. Павловская Т.А., Щупак Ю.А. С++. Объектно-ориентированное программирование: Практикум. – СПб.: Питер,2004 -288 с.
3. Павловская Т.А.С/С++ Программирование на языке высокого уровня. – СПб.: Питер,2007 -461 с.
4. Х.М.Дейтел., П.Дж.Дейтел. Как программировать на Си++. М .: ЗАО БИНОМ, 1999, 1000 с.
5. Карпов Б., Баранов Т. С++: специальный практикум. – СПб.: Питер,2000 -480 с.
6. Лаптев В.В. C++. Экспресс-курс. – СПб.: БХВ-Петербург, 2004. – 512 с.: ил.
7. Лафоре Р. Объектно-ориентированное программирование в С++. Классика Computer Science. 4 изд.. – СПб.: Питер, 2003.

**ДОДАТКИ**

**Додаток 1. Текст програми**

#include <iostream>

Изм

Аркуш

№ докум

Подп

Дата

Арк

#include "Date.h"

#include "ValidationException.h"

#include "Order.h"

#include "DailyReport.h"

using namespace std;

const int arrSize = 10;

string inputFilename() {

string filename;

while (true) {

cin.clear();

cin.sync();

cout << "Enter filename:" << endl;

getline(cin, filename);

if (filename.empty()) {

cerr << "Wrong filename" << endl;

continue;

}

return filename;

}

}

template<class T>

T averageProductsCount(Array<Order \*> orders) { // Середня кількість товарів

T sum = 0;

for (int i = 0; i < orders.size(); ++i) {

sum += orders[i]->getProducts().size();

}

return sum / orders.size();

}

template<typename T>

T averageTotal(Array<Order \*> orders) { // Середня сума товарів

T sum = 0;

for (int i = 0; i < orders.size(); ++i) {

sum += orders[i]->getTotalPrice();

}

return sum / orders.size();

}

void handleCmd(const char cmd, DailyReport \*dailyReport) {

cout << endl;

switch (cmd) {

case '1': // Заповнення об’єктів

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

cout << "Filling Order #" << i + 1 << " of " << arrSize << endl;

auto order = new Order();

auto customer = new Customer();

Изм

Аркуш

№ докум

Подп

Дата

Арк

cin >> customer;

cout << endl;

order->setCustomer(customer);

for (int j = 0; j < arrSize; ++j) {

cout << "Filling Product #" << j + 1 << " of " << arrSize << endl;

auto product = new Product();

cin >> product;

cout << endl;

order->addProduct(product);

}

order->setDate(dailyReport->getDate());

dailyReport->addOrder(order);

}

break;

case '2': // Виведення у табличному виді

cout << dailyReport << endl;

break;

case '3': // Збереження у файл

char choice;

do {

string filename;

cout << "Select file type:" << endl;

cout << "|1| Binary\n";

cout << "|2| Text\n";

cout << "|0| Back\n";

cout << "Enter option number:" << endl;

cin >> choice;

if (choice == '1' || choice == '2') {

filename = inputFilename();

ofstream file(filename, ios::binary);

if (!file.is\_open()) {

cerr << "Cannot open file" << endl;

continue;

}

if (choice == '1') {

unsigned int size = dailyReport->getOrders().size();

file.write((char \*) &size, sizeof(unsigned int));

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

(\*dailyReport)[i]->write(file);

}

} else if (choice == '2') {

file << dailyReport->jsonSerialize().dump(4);

}

file.close();

break;

}

} while (choice != '0');

break;

case '4': { // Зчитування з файлу

string filename = inputFilename();

ifstream file(filename, ios::binary);

Изм

Аркуш

№ докум

Подп

Дата

Арк

if (!file.is\_open()) {

cerr << "Cannot open file" << endl;

break;

}

do {

cout << "Select file type:" << endl;

cout << "|1| Binary\n";

cout << "|2| Text\n";

cout << "|0| Back\n";

cout << "Enter option number:" << endl;

cin >> choice;

dailyReport->clear();

if (choice == '1') { // Обрано бінарний файл

int size;

file.read((char \*) &size, sizeof(int));

for (size\_t i = 0; i < size; i++) {

auto order = new Order();

order->read(file);

dailyReport->addOrder(order);

delete order;

}

cout << "Success!" << endl;

break;

} else if (choice == '2') { // Обрано текстовий файл

dailyReport->jsonDeserialize(file);

cout << "Success!" << endl;

break;

}

} while (choice != '0');

file.close();

break;

}

case '5': { // Пошук за полями

do {

cout << "Select search type:" << endl;

cout << "|1| By text\n";

cout << "|2| By number\n";

cout << "|0| Back\n";

cout << "Enter option number:" << endl;

cin >> choice;

cin.clear();

cin.sync();

if (choice == '1') { // Текстове поле

string str;

getline(cin, str);

dailyReport->searchByText(str);

choice = '0'; // Вихід з циклу

} else if (choice == '2') { // Числове поле

int look;

cin >> look;

dailyReport->searchByNum(look);

choice = '0'; // Вихід з циклу

}

} while (choice != '0');

break;

}

Изм

Аркуш

№ докум

Подп

Дата

Арк

case '6': { // Розраухнок статистичних даних

cout << "Products count: " << averageProductsCount<float>(dailyReport->getOrders()) << endl;

cout << "Average total: " << averageTotal<float>(dailyReport->getOrders()) << endl;

break;

}

default:

cerr << "Invalid option number" << endl;

break;

}

cout << "\n";

}

int main() {

auto dailyReport = new DailyReport();

dailyReport->setDate(Date(2019, 12, 22));

char option;

do { // Меню

cout << "Menu:\n\n";

cout << "|1| Create Objects\n";

cout << "|2| Print data\n";

cout << "|3| Write to file\n";

cout << "|4| Read from file\n";

cout << "|5| Search\n";

cout << "|6| Show average\n";

cout << "|0| Exit\n";

cout << "Enter option number:" << endl;

cin >> option;

try {

handleCmd(option, dailyReport);

} catch (ValidationException &e) { // Обробка виключень

cerr << e.what() << endl;

continue;

} catch (exception &e) {

cerr << "Unexpected error, interrupting: " << e.what() << endl;

break;

}

} while (option != '0');

}

**Додаток 2. Опис програми**

**2.1 Загальні відомості**

Изм

Аркуш

№ докум

Подп

Дата

Арк

Програма була розроблена на мові програмування С++ з використанням об’єктно орієнтованого стилю програмування у середовищі розробки CLion, компілятору g++ та CMake.

Вихідним файлом компіляції є .exe файл для операційної системи Windows та виконавчий файл для UNIX.

Спілкування користувача з програми відбувається після її запуску через консоль. Виведення даних у табличному вигляді.

**2.2 Функціональне призначення**

Програма створена для заповнення даними об’єктів реального світу, зокрема має такий функціонал:

* Зручне меню для керування програмою
* Введення даних з клавіатури
* Виведення даних у табличному вигляді
* Зчитування та запис у бінарний файл
* Зчитування та запис у текстовий файл, зокрема у форматі JSON (JavaScript Object Notation)
* Пошук серед текстових та числових значень
* Розрахунок статистичних даних (середня кількість товарів, середня сума товарів)

**2.3 Опис логічної структури**

Изм

Аркуш

№ докум

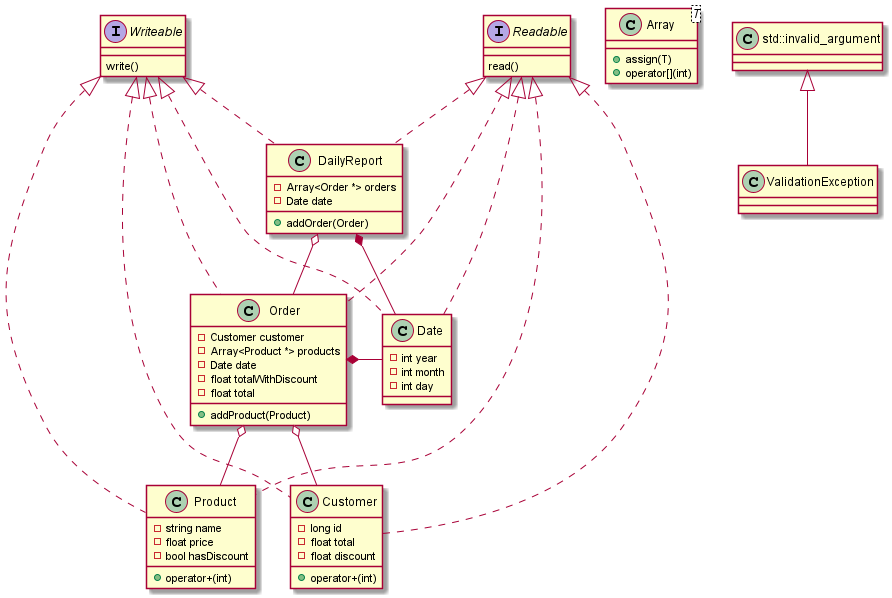
Подп

Дата

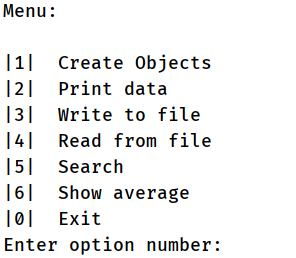
Арк

Код програми складається з файлів .h (оголошення класів) та .cpp (реалізація класів).

Структура класів наведена у UML діаграмі 2.3.1 (також див. розділ 1 основної частини).



Діаграма 2.3.1 UML діаграма структури класів



Изм

Аркуш

№ докум

Подп

Дата

Арк

Рисунок 2.3.2 Структура Меню

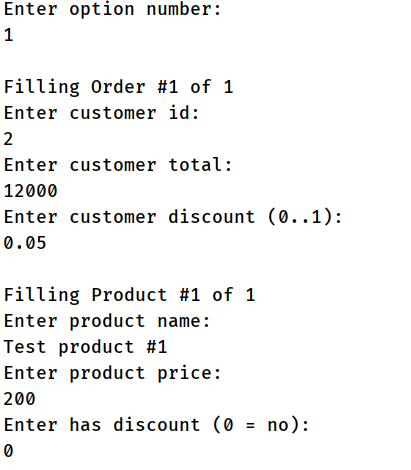


Рисунок 2.3.3 Заповнення з клавіатури

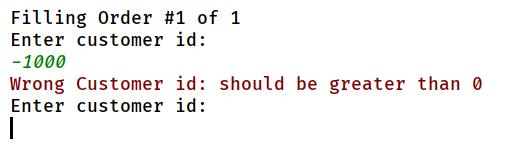
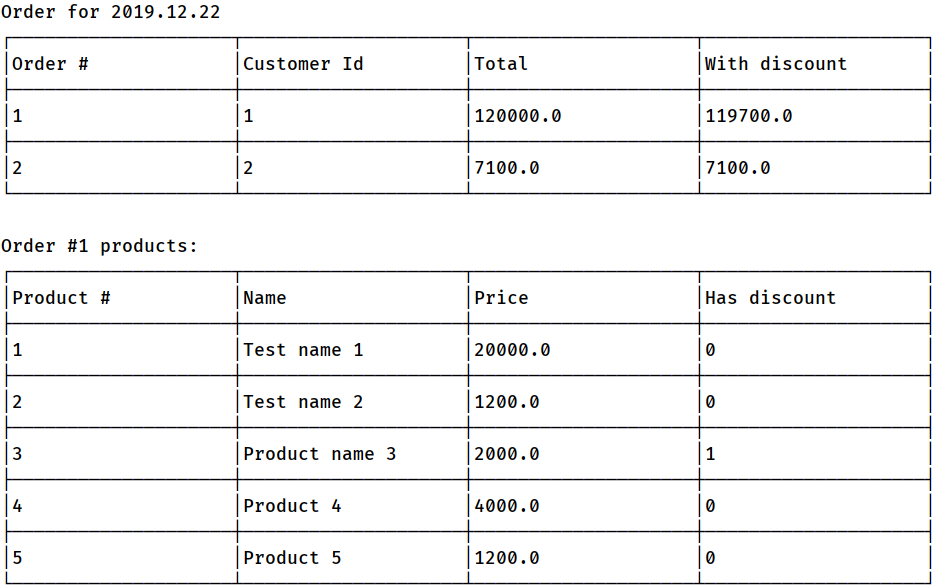


Рисунок 2.3.4 Приклад обробки виключень



Изм

Аркуш

№ докум

Подп

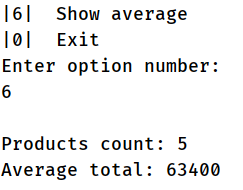
Дата

Арк

Рисунок 2.3.5 Виведення у вигляді таблиці



Рисунок 2.3.6 Структура текстового файлу у форматі json



Изм

Аркуш

№ докум

Подп

Дата

Арк

Рисунок 2.3.7 Виведення середніх значень

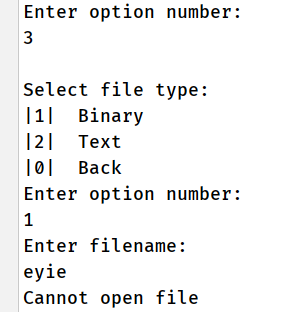


Рисунок 2.3.8 Приклад обробки виключень

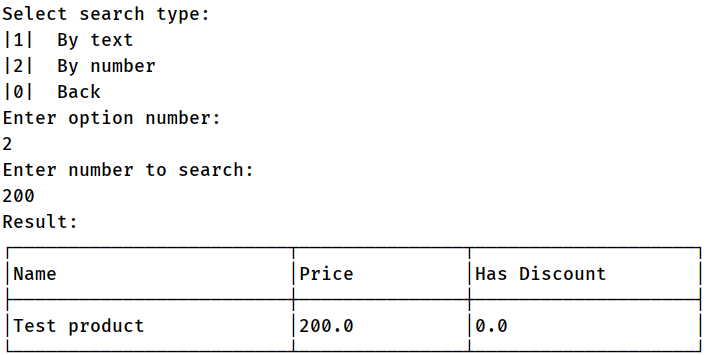


Рисунок 2.3.9 Приклад пошуку за числом