**Документация за курсов проект по ПЕ  
  
*“Приложение за управление на доставчици”  
(Supplier Management App)***

Изготвил: Марин Стоянов, 121222169, гр. 45А

**Задание**

Създайте приложение, което да поддържа информация за доставчици на различни материали от оптиката. Работи се с доставчици, за които трябва да се съхранява техния Булстат, името и местонахождението им и телефонен номер. Всеки доставчик може да достави един или повече оптика, които се характеризират с тип, дебелина, диоптър и наименованието на материала.  
  
Приложението да има възможност за въвеждане на произволен брой различни доставчици и оптика.  
  
Да има възможност за избор на доставчик и от него да се поръчват различни материали, като се изчислява общата цена на поръчката.  
Класовете (най-малко 3 класа при реализацията) трябва да капсулира всичките детайли. Използват се private инстанции на променливите за съхраняване на различните детайли. Трябва да има най-малко два конструктора, public getters/setters за private инстанции на променливите.  
  
Необходимо е да извършвате проверка на входните данни .  
Да се предефинира операцията <<, която да се използва за извеждане на данните . Данните да се четат и съхраняват във файл .

Класовете да се опишат с UML клас диаграма.  
Задължително данните да се въвеждат динамично, чрез меню.

**Общ преглед**

Приложението за управление на доставчиците е софтуер на C++, предназначен за управление на доставчиците и свързаните с тях материали, като предоставя функционалности за добавяне, промяна и премахване на доставчици, както и за пазаруване на оптични материали от изброените доставчици.

**Характеристики**

1. **Управление на доставчиците:**

* Добавяне на нови доставчици.
* Промяна на данните на доставчик.
* Премахване на съществуващи доставчици.
* Списък на всички доставчици.

1. **Управление на материалите:**

* Добавяне на продукти към доставчик.
* Преглед на продуктите на доставчика.

1. **Пазаруване:**

* Избиране на доставчик и преглеждане на неговите продукти.
* Поръчка на материали от даден доставчик и изчисляване на общата цена.

1. **Постоянство на данните:**

* Записване на данни за доставчиците и продуктите във файл
* Зареждане на данни за доставчици и продукти от файл.

**Описание на класовете**

**Картина, която съдържа текст, екранна снимка, Шрифт

Описанието е генерирано автоматично**

**SupplierManagement**

*Header file: SupplierManagement.hpp  
Implementation file: SupplierManagement.cpp*

**Отговорности:**

* Действа като основен интерфейс за работа с приложението.
* Обслужва взаимодействието с потребителите за управление на доставчиците и материалите.
* Връзка с базата данни за запазване на данните.

**Основни методи:**

void Run(): Основен метод за стартиране на приложението.

void addSupplier(const Supplier& supplier): Добавя нов доставчик.

void removeSupplier(const int& index): Премахва доставчик по индекс.

void modifySupplier(const int& index): Променя данните на доставчик.

void listSuppliers() const: Изброява всички доставчици.

void showShoppingMenu(): Обработва менюто за пазаруване.

void showManagerMenu(): Обработва менюто на мениджъра.

void loadDatabase(): Зарежда данни за доставчика от базата данни.

void saveDatabase(): Записва данните за доставчика в базата данни.

**Supplier**

*Header file: Supplier.hpp  
Implementation file: Supplier.cpp*

**Отговорности:**

* Представлява доставчик със съответните детайли и материали.
* Предоставя методи за управление на материалите.

**Ключови атрибути:**

std::string m\_Bulstat: Уникален идентификатор на доставчика.

std::string m\_Name: Име на доставчик

std::string m\_Location: Локация на доставчик

std::string m\_Phone: Телефонен номер на доставчик

std::vector<OpticMaterial> m\_Materials: Списък на продуктите, предоставени от доставчика.

**Основни методи:**

void addMaterial(const OpticMaterial& material): Добавя продукт към даден доставчик.

void listMaterials() const: Извежда всички продукти за даден доставчик.

float orderMaterials(const std::vector<int>& materialIndexes) const: Пресмята крайната цена на избраните продукти.

**OpticMaterial**

*Header file: OpticsMaterial.hpp  
Implementation file: OpticsMaterial.cpp*

**Отговорности:**

* Представлява оптичен материал с атрибути като тип, дебелина, диоптър, материал и цена.

**Ключови атрибути:**

std::string m\_Type: Тип на оптичния продукт.

float m\_Thickness: Дебелина в милиметри.

float m\_Diopter: Стойност на оптичния диоптър.

std::string m\_Material: Тип материал.

float m\_Price: Цена на продукт.

**Основни методи:**

Getters: Извличане на отделни атрибути (getType(), getMaterial() и т.н.).

Overloaded operator<<: Извеждане на данни за материала във форматиран вид.

**Utility**

*Header file: Utility.hpp*

**Отговорности:**

* Осигурява статични помощни функции за общи операции и управление на менюто.

**Основни методи:**

int MainMenuGetChoice(): Показва и извлича избора на главното меню.

int ShoppingMenuGetChoice(): Извлича избора от менюто за пазаруване.

int ManagerMenuGetChoice(int& choice): Показва и извлича избора на менюто на мениджъра.

bool AddProducts(): Запитва дали да се добавят продукти към доставчик.

void trimWhitespaces(std::string& s): Премахва началните и крайните бели полета от низ.

**Допълнение:**

enum MENU\_TYPE: Опции за главно меню.

enum MANAGER\_MENU\_OPTIONS: Опции за „мениджърско“ меню.

static const char MAIN\_MENU[],  
static const char SUPPLIER\_MENU[],  
static const char PRODUCT\_CHOICE[]: Спомагателни масиви от символи за извеждане на различни менюта.

static const char DATABASE[]: Път към „базата от данни“.