Картина, която съдържа текст, Шрифт, лого, символ

Описанието е генерирано автоматично

**Курсова проект по  
Програмни езици**

Оптични материали

Изготвил: Кристиян Стойков  
Специалност: КСИ  
Фак. номер: 121221086  
Група: 42

Съдържание

[Оптични материали 3](#_Toc153343730)

[Инсталация 3](#_Toc153343731)

[UML диаграма 3](#_Toc153343732)

[Класове 3](#_Toc153343733)

[Base (base.h) 3](#_Toc153343734)

[Optic\_Material (optic\_material.h) 4](#_Toc153343735)

[Optic\_Materials (optic\_materials.h) 5](#_Toc153343736)

[Order (order.h) 5](#_Toc153343737)

[Orders (orders.h) 6](#_Toc153343738)

[Supplier (supplier.h) 6](#_Toc153343739)

[Suppliers (suppliers.h) 7](#_Toc153343740)

[Main (main.cpp) 8](#_Toc153343741)

# Оптични материали

Оптични материали е курсов проект по "Програмни езици" от 3ти курс 1ви семестър.

Проекта е написан на: c++.

## Инсталация

За да стартирате "Оптични материали", ще трябва да имате инсталиран Visual Studio 2022 или VS Code или компилатор за c++. Освен това за този проект е необходима библиотеката **nlohmann/json**. Можете да инсталирате тази библиотека като следвате инструкциите на техния [GitHub](https://github.com/nlohmann/json).

След като инсталирате необходимия софтуер и библиотеки, можете да клонирате това хранилище с помощта на следната команда:

git clone https://github.com/kristiyanstoykov/optic\_materials.git

## UML диаграма

Картина, която съдържа текст, диаграма, План, линия

Описанието е генерирано автоматично

## Класове

### Base (base.h)

Базов клас, осигуряващ интерфейс за сериализация и десериализация, както и за принтиране и въвеждане.

#### **Функции**

* virtual ostream& print(ostream& output) const: Виртуален метод за отпечатване на дaнни.
* virtual istream& input(istream& input): Виртуален метод за четене на данни.
* virtual void to\_json(json&j) const: Виртуален метод за записване в JSON.
* virtual void from\_json(json&j): Виртуален метод за четене на обекта от JSON.
* friend ostream& operator<<(ostream& output, const Base& base): Предефиниране на оператора << за извеждане на данните.
* friend istream& operator>>(istream& input, Base& base): Предефиниране на оператора >> за четене на данни.

### Optic\_Material (optic\_material.h)

Разширява базовия клас. Представлява оптичен материал и управлява неговите свойства.

#### **Променливи на класа**

* string type: Тип на оптичния материал.
* double width: Ширина на оптичния материал.
* double diopter: Диоптърна стойност на оптичния материал.
* string name: Име на оптичния материал.
* double price: Цена на оптичния материал.

#### **Конструктори**

* Optic\_Material(): Конструктор по подразбиране.
* Optic\_Material(string type, double width, double diopter, string name, double price): Конструктор с параметри за тип, ширина, диоптър, име и цена.

#### **Методи на класа**

* string getType() const: Връща типа на оптичния материал.
* void setType(string type): Задава типа на оптичния материал.
* double getWidth() const: Връща ширината на оптичния материал.
* void setWidth (double width): Задава ширината на оптичния материал.
* double getDiopter() const: Връща стойността на диоптъра на оптичния материал.
* void setDiopter (double diopter): Задава стойността на диоптъра на оптичния материал.
* string getName() const: Връща името на оптичния материал.
* void setName (string name): Задава името на оптичния материал.
* double getPrice() const: Връща цената на оптичния материал.
* void setPrice (double price): Определя цената на оптичния материал.

### Optic\_Materials (optic\_materials.h)

Разширява базовия клас. Управлява колекция от Optic\_Material обекти с функционалности за обработка на колекцията и сериализация/десериализация.

#### **Променливи на класа**

* vector<Optic\_Material> optic\_materials: Колекция от Optic\_Material обекти.

#### **Конструктори**

* Optic\_Materials(): Конструктор по подразбиране.

#### **Методи на класа**

* int getSize() const: Връща броя на оптичните материали в колекцията.
* bool isEmpty(): Проверява дали колекцията е празна.
* void addOpticMaterial(const Optic\_Material & optic\_material): Добавя Optic\_Material обект към колекцията.
* vector<Optic\_Material> getOpticMaterials(): Връща всички оптични материали в колекцията.
* Optic\_Material getOpticMaterialByIndex(int index) const: Връща Optic\_Material обект при определен индекс в колекцията.
* void print\_on\_one\_line() const: Отпечатва едноредово описание на всеки оптичен материал в колекцията.

### Order (order.h)

Разширява базовия клас. Представлява поръчка, обхващаща оптични материали и доставчик, с функционалности за управление на поръчката и изчисляване на общата й стойност.

#### **Променливи**

* int id: Идентификатор на поръчката.
* vector<Optic\_Material> materials: Списък на оптичните материали, включени в поръчката.
* Supplier supplier: Доставчикът, свързан с поръчката.

#### **Конструктори**

* Order(): Конструктор по подразбиране.
* Order (int id, vector<Optic\_Material> materials, Supplier supplier): Конструктор с параметри за номер на поръчката, материалите и доставчика.

#### **Методи на класа**

* void setId(int id): Задава номер на поръчката.
* int getId(): Получава уникалния номер на поръчката.
* vector<Optic\_Material> getMaterials(): Взима всички оптични материали, включени в поръчката.
* void addMaterial(const Optic\_Material& material): Добавя оптичен материал към поръчката.
* void addSupplier(const Supplier& supplier): Задава доставчика за поръчката.
* Доставчик getSupplier(): Взима доставчика, свързан с поръчката.
* double getTotalRaw(): Изчислява общата цена на поръчката без процента печалбата на доставчика.
* double getTotal(): Изчислява общата цена на поръчката, включително процента на печалба на доставчика.

### Orders (orders.h)

Разширява базовия клас. Управлява колекция от обекти за поръчка с функционалности за обработка и управление на колекцията и сериализация/десериализация.

#### **Променливи на класа**

* vector<Order> orders: Колекция от обекти поръчка .

#### **Конструктори**

* Orders(): Конструктор по подразбиране.

#### **Методи на класа**

* void addOrder(const Order& order): Добавя обект Order към колекцията.
* vector<Order> getOrders(): Взима всички поръчки в колекцията.
* void addMaterialToLastOrder (const Optic\_Material& materials): Добавя оптичен материал към последната поръчка в колекцията.
* void addSupplierToLastOrder (const Supplier& supplier): Добавя доставчик към последната поръчка в колекцията.
* void addIdToLastOrder(int id): Задава ID за последната поръчка в колекцията.
* void printOrderTotal(): Отпечатва общата стойност на всяка поръчка в колекцията.

### 

### Supplier (supplier.h)

Разширява базовия клас. Представлява доставчик в системата за оптични материали с подробности за идентификация, информация за контакт и марж на печалба.

#### **Променливи на класа**

* string bulstat:Булстат за доставчика.
* string name: Име на доставчика.
* string location: Местоположение на доставчика.
* string phone: Телефонен номер на доставчика.
* double profit\_margin: Печалба на доставчика в процент.

#### **Конструктори**

* Supplier(): Конструктор по подразбиране.
* Supplier (string bulstat, string name, string location, string phone, double profit\_margin): Конструктор с параметри за булстат, име, местоположение, телефон и печалба.

#### **Методи на класа**

* std::string getBulstat() const: Получава булстата на доставчика.
* void setBulstat(string bulstat): Задава булстата на доставчика.
* std::string getName() const: Получава името на доставчика.
* void setName(string name): Задава името на доставчика.
* std::string getLocation() const: Получава местоположението на доставчика.
* void setLocation(string location): Задава местоположението на доставчика.
* std::string getPhone() const: Получава телефонния номер на доставчика.
* void setPhone(string phone): Задава телефонния номер на доставчика.
* double getProfitMargin () const: Получава печалбата на доставчика.
* void setProfitMargin(double profit\_margin): Задава печалбата на доставчика.

### Suppliers (suppliers.h)

Разширява базовия клас. Управлява колекция от обекти на доставчика с функционалности за добавяне и достъп до доставчици и внедряване на сериализация/десериализация.

#### **Променливи на класа**

* vector<supplier> suppliers: Колекция от обекти на доставчика .

#### **Конструктори**

* Suppliers(): Конструктор по подразбиране.

#### **Методи на класа**

* void addSupplier(const Supplier& supplier): Добавя обект на доставчик към колекцията.
* bool isEmpty(): Проверява дали колекцията е празна.
* vector<Supplier> getSuppliers(): Извлича всички доставчици в колекцията.
* Suppliers getSupplierByIndex(int index) const: Извлича обект на доставчик на определен индекс в колекцията.
* int getSize() const: Получава броя на доставчиците в колекцията.
* void print\_on\_one\_line() const: Отпечатва едноредово описание на всеки доставчик в колекцията.

### Main (main.cpp)

* load\_orders(Orders &orders) Зарежда поръчки от текстов файл в обекта Поръчки.
* load\_orders\_json(Orders & orders) Зарежда поръчки от JSON файл в обекта Orders.
* save\_orders(Orders orders) Записва обекта Поръчки в текстов файл.
* save\_orders\_json(Orders orders) Записва обекта Orders в JSON файл.
* load\_suppliers(Suppliers &suppliers) Зарежда доставчици от текстов файл в обекта на Доставчиците .
* load\_suppliers\_json(Suppliers &suppliers) Зарежда доставчици от JSON файл в обекта на Доставчиците .
* save\_suppliers(Suppliers suppliers) Запазва Доставчиците възразяват срещу текстов файл.
* save\_suppliers\_json (Suppliers suppliers) Запазва Доставчиците в JSON файл.
* load\_materials(Optic\_Materials & materials) Зарежда материали от текстов файл в Optic\_Materials обект.
* load\_materials\_json(Optic\_Materials & materials) Зарежда материали от JSON файл в Optic\_Materials обект.
* save\_materials(Optic\_Materials materials) Записва текущото състояние на Optic\_Materials обект в текстов файл.
* save\_materials\_json(Optic\_Materials materials) Записва текущото състояние на Optic\_Materials обект в JSON файл.
* enter\_supplier() Въвеждане на данни за доставчика и връща обект на доставчика.
* enter\_material() Подканва потребителя да въведе материал и връща Optic\_Material обект.
* enter\_order(Orders &orders, Optic\_Materials materials, Suppliers suppliers) Позволява на потребителя да създаде нова поръчка и да я добави към Поръчки.
* display\_menu(Orders &orders, Suppliers &suppliers, Optic\_Materials &optic\_materials) Показва меню на потребителя и обработва въведените от потребителя данни за различни действия.