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

[Optic Materials 3](#_Toc153293366)

[Installation 3](#_Toc153293367)

[Usage 3](#_Toc153293368)

[Features 3](#_Toc153293369)

[Base (base.h) 3](#_Toc153293370)

[Optic\_Material (optic\_material.h) 4](#_Toc153293371)

[Optic\_Materials (optic\_materials.h) 5](#_Toc153293372)

[Order (order.h) 5](#_Toc153293373)

[Orders (orders.h) 6](#_Toc153293374)

[Supplier (supplier.h) 7](#_Toc153293375)

[Suppliers (suppliers.h) 8](#_Toc153293376)

[Main (main.cpp) 8](#_Toc153293377)

[Contact 9](#_Toc153293378)

# Optic Materials

Optic Materials is a course project for the Programming Languages subject.

Made with: c++.

## Installation

To get started with “Optic Materials”, you’ll need to have either Visual Studio 2022 or VS Code installed on your system. Additionally, the **nlohmann/json library** is required for this project. You can install this library via your preferred package manager or by following the instructions on its [GitHub repository](https://github.com/nlohmann/json).

## Usage

After installing the required software and libraries, you can clone this repository using the following command:

Once cloned, open the project in Visual Studio 2022 or VS Code and run the application.

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

## Features

### Base (base.h)

A base class providing an interface for serialization and deserialization, as well as stream insertion and extraction.

#### **Functions**

* virtual ostream& print(ostream& output) const: Virtual method for printing the object’s details to an output stream.
* virtual istream& input(istream& input): Virtual method for reading data into the object from an input stream.
* virtual void to\_json(json& j) const: Virtual method for converting the object’s state to JSON format.
* virtual void from\_json(json& j): Virtual method for setting the object’s state from JSON format.
* friend ostream& operator<<(ostream& output, const Base& base): Overloads the << operator for outputting the Base object’s details to an output stream.
* friend istream& operator>>(istream& input, Base& base): Overloads the >> operator for reading data into a Base object from an input stream.

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

Extends the Base class. Represents an optical material and manages its properties.

#### **Member Variables**

* string type: Type of the optical material.
* double width: Width of the optical material.
* double diopter: Diopter value of the optical material.
* string name: Name of the optical material.
* double price: Price of the optical material.

#### **Constructors**

* Optic\_Material(): Default constructor.
* Optic\_Material(string type, double width, double diopter, string name, double price): Constructor with parameters for type, width, diopter, name, and price.

#### **Member Functions**

* string getType() const: Returns the type of the optical material.
* void setType(string type): Sets the type of the optical material.
* double getWidth() const: Returns the width of the optical material.
* void setWidth(double width): Sets the width of the optical material.
* double getDiopter() const: Returns the diopter value of the optical material.
* void setDiopter(double diopter): Sets the diopter value of the optical material.
* string getName() const: Returns the name of the optical material.
* void setName(string name): Sets the name of the optical material.
* double getPrice() const: Returns the price of the optical material.
* void setPrice(double price): Sets the price of the optical material.

#### **Overrides from Base**

ostream& print(ostream& output) const: Prints the optical material’s details to an output stream.

istream& input(istream& input): Reads data into the optical material from an input stream.

void to\_json(json& j) const: Converts the optical material’s state to JSON format.

void from\_json(json& j): Sets the optical material’s state from JSON format.

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

Extends the Base class. Manages a collection of Optic\_Material objects with functionalities for handling the collection and serialization/deserialization.

#### **Member Variables**

* vector<Optic\_Material> optic\_materials: A collection of Optic\_Material objects.

#### **Constructors**

* Optic\_Materials(): Default constructor.

#### **Member Functions**

* int getSize() const: Returns the number of optic materials in the collection.
* void addOpticMaterial(const Optic\_Material& optic\_material): Adds an Optic\_Material object to the collection.
* vector<Optic\_Material> getOpticMaterials(): Retrieves all optic materials in the collection.
* Optic\_Material getOpticMaterialByIndex(int index) const: Retrieves an Optic\_Material object at a specified index in the collection.
* void print\_on\_one\_line() const: Prints a concise one-line description of each optic material in the collection.

#### **Overrides from Base**

* ostream& print(ostream& output) const: Prints the collection’s details to an output stream.
* istream& input(istream& input): Reads data into the collection from an input stream.
* void to\_json(json& j) const: Converts the collection’s state to JSON format.
* void from\_json(json& j): Sets the collection’s state from JSON format.

### Order (order.h)

Extends the Base class. Represents an order, encompassing optic materials and a supplier, with functionalities for managing the order and calculating its total cost.

#### **Member Variables**

* int id: Unique identifier for the order.
* vector<Optic\_Material> materials: A list of optic materials included in the order.
* Supplier supplier: The supplier associated with the order.

#### **Constructors**

* Order(): Default constructor.
* Order(int id, vector<Optic\_Material> materials, Supplier supplier): Constructor with parameters for the order’s ID, materials, and supplier.

#### **Member Functions**

* void setId(int id): Sets the unique identifier of the order.
* int getId(): Gets the unique identifier of the order.
* vector<Optic\_Material> getMaterials(): Retrieves all optic materials included in the order.
* void addMaterial(const Optic\_Material& material): Adds an optic material to the order.
* void addSupplier(const Supplier& supplier): Sets the supplier for the order.
* Supplier getSupplier(): Gets the supplier associated with the order.
* double getTotalRaw(): Calculates the total cost of the order without profit margin.
* double getTotal(): Calculates the total cost of the order including the supplier’s profit margin.

#### **Overrides from Base**

* ostream& print(ostream& output) const: Prints the order’s details to an output stream.
* istream& input(istream& input): Reads data into the order from an input stream.
* void to\_json(json& j) const: Converts the order’s state to JSON format.
* void from\_json(json& j): Sets the order’s state from JSON format.

### Orders (orders.h)

Extends the Base class. Manages a collection of Order objects with functionalities for handling and managing the collection and serialization/deserialization.

#### **Member Variables**

* vector<Order> orders: A collection of Order objects.

#### **Constructors**

* Orders(): Default constructor.

#### **Member Functions**

* void addOrder(const Order& order): Adds an Order object to the collection.
* vector<Order> getOrders(): Retrieves all orders in the collection.
* void addMaterialToLastOrder(const Optic\_Material& material): Adds an optic material to the last order in the collection.
* void addSupplierToLastOrder(const Supplier& supplier): Adds a supplier to the last order in the collection.
* void addIdToLastOrder(int id): Sets the ID for the last order in the collection.
* void printOrderTotal(): Prints the total cost of each order in the collection.

#### **Overrides from Base**

* ostream& print(ostream& output) const: Prints the collection’s details to an output stream.
* istream& input(istream& input): Reads data into the collection from an input stream.
* void to\_json(json& j) const: Converts the collection’s state to JSON format.
* void from\_json(json& j): Sets the collection’s state from JSON format.

### Supplier (supplier.h)

Extends the Base class. Represents a supplier in the optics materials system with details about identification, contact information, and profit margin.

#### **Member Variables**

* std::string bulstat: Unique identifier for the supplier.
* std::string name: Name of the supplier.
* std::string location: Location of the supplier.
* std::string phone: Contact phone number of the supplier.
* double profit\_margin: Profit margin of the supplier.

#### **Constructors**

* Supplier(): Default constructor.
* Supplier(std::string bulstat, std::string name, std::string location, std::string phone, double profit\_margin): Constructor with parameters for bulstat, name, location, phone, and profit margin.

#### **Member Functions**

* std::string getBulstat() const: Gets the unique identifier of the supplier.
* void setBulstat(std::string bulstat): Sets the unique identifier of the supplier.
* std::string getName() const: Gets the name of the supplier.
* void setName(std::string name): Sets the name of the supplier.
* std::string getLocation() const: Gets the location of the supplier.
* void setLocation(std::string location): Sets the location of the supplier.
* std::string getPhone() const: Gets the contact phone number of the supplier.
* void setPhone(std::string phone): Sets the contact phone number of the supplier.
* double getProfitMargin() const: Gets the profit margin of the supplier.
* void setProfitMargin(double profit\_margin): Sets the profit margin for the supplier.

#### **Overrides from Base**

* ostream& print(ostream& output) const: Prints the supplier’s details to an output stream.
* istream& input(istream& input): Reads data into the supplier from an input stream.
* void to\_json(nlohmann::json& j) const: Converts the supplier’s state to JSON format.
* void from\_json(json& j): Sets the supplier’s state from JSON format.

### Suppliers (suppliers.h)

Extends the Base class. Manages a collection of Supplier objects with functionalities for adding and accessing suppliers, and implementing serialization/deserialization.

#### **Member Variables**

* vector<Supplier> suppliers: A collection of Supplier objects.

#### **Constructors**

* Suppliers(): Default constructor.

#### **Member Functions**

* void addSupplier(const Supplier& supplier): Adds a Supplier object to the collection.
* vector<Supplier> getSuppliers(): Retrieves all suppliers in the collection.
* Supplier getSupplierByIndex(int index) const: Retrieves a Supplier object at a specified index in the collection.
* int getSize() const: Gets the number of suppliers in the collection.
* void print\_on\_one\_line() const: Prints a concise one-line description of each supplier in the collection.

#### **Overrides from Base**

* ostream& print(ostream& output) const: Prints the collection’s details to an output stream.
* istream& input(istream& input): Reads data into the collection from an input stream.
* void to\_json(json& j) const: Converts the collection’s state to JSON format.
* void from\_json(json& j): Sets the collection’s state from JSON format.

### Main (main.cpp)

* load\_orders(Orders &orders) Loads orders from a text file into the Orders object.
* load\_orders\_json(Orders &orders) Loads orders from a JSON file into the Orders object.
* save\_orders(Orders orders) Saves the current state of Orders object to a text file.
* save\_orders\_json(Orders orders) Saves the current state of Orders object to a JSON file.
* load\_suppliers(Suppliers &suppliers) Loads suppliers from a text file into the Suppliers object.
* load\_suppliers\_json(Suppliers &suppliers) Loads suppliers from a JSON file into the Suppliers object.
* save\_suppliers(Suppliers suppliers) Saves the current state of Suppliers object to a text file.
* save\_suppliers\_json(Suppliers suppliers) Saves the current state of Suppliers object to a JSON file.
* load\_materials(Optic\_Materials &materials) Loads materials from a text file into the Optic\_Materials object.
* load\_materials\_json(Optic\_Materials &materials) Loads materials from a JSON file into the Optic\_Materials object.
* save\_materials(Optic\_Materials materials) Saves the current state of Optic\_Materials object to a text file.
* save\_materials\_json(Optic\_Materials materials) Saves the current state of Optic\_Materials object to a JSON file.
* enter\_supplier() Prompts the user to enter supplier details and returns a Supplier object.
* enter\_material() Prompts the user to enter material details and returns an Optic\_Material object.
* enter\_order(Orders &orders, Optic\_Materials materials, Suppliers suppliers) Allows the user to create a new order and add it to Orders.
* display\_menu(Orders &orders, Suppliers &suppliers, Optic\_Materials &materials) Displays a menu to the user and handles user input for various actions.
* test(Suppliers &suppliers) A test function to add sample suppliers to the Suppliers object.

## Contact

Email: [kstoykov@tu-sofia.bg](mailto:kstoykov@tu-sofia.bg)