**DOCUMENTATION. PROJECT PHARMACY**

**The essence of the project**

The pharmacy is the main facility, which includes a warehouse of medicines, a seller and a client. Communication is set up in such a way that first the seller asks about the medicine that the client wants to buy. If the medicine is on prescription, the seller asks for the prescription.Further, if this product is in stock and its quantity is sufficient according to the customer's requirement, you can buy it.If this product is either not there, or it is not enough, you can order it and after some time it will be available.

It is also possible to buy and order several medicines, as well as change the user an unlimited number of times.

* **PharmacyShop.h**

This header file defines the PharmacyShop template class, which represents a pharmacy shop. It contains the following members:

- name: A string representing the name of the pharmacy shop.

- storage: An instance of the Storage class template, representing the storage of products in the shop.

- seller: An instance of the Seller class, representing the seller in the shop.

The PharmacyShop class provides the following member functions:

- PharmacyShop(Storage<Product> storage, Seller seller, std::string name): A constructor that initializes the storage, seller, and name members with the provided arguments.

- greeting(): Prints a welcome message to the customer, including the sellers information and the shops name.

- askForPurchase(Customer& customer): Prompts the customer to specify what they want to buy.

- isAvailable(std::string name): Checks if a product with the given name is available in the shops storage.

- getProduct(std::string name): Retrieves the product with the given name from the shops storage.

- printInformationAboutDrug(Product product): Prints information about the given product.

- needPrescription(Product product): Checks if the given product requires a prescription.

- askForPrescription(): Prompts the customer to provide information about whether they have a prescription for a drug.

- orderProduct(Product product): Places an order for the given product, setting a random waiting time for its availability.

- sellProduct(Product product, int quantity\_sold, Customer& customer): Sells the specified quantity of the given product to the customer, updating the storage and adding the cost to the customers receipt.

* **Storage.h**

This header file defines the Storage template class, which represents the storage of products in the pharmacy shop. It manages the availability, quantity, and waiting time of products. The class contains the following members:

- products: A vector of pairs, where each pair consists of a product of type Product and its quantity.

The Storage class provides the following member functions:

- Storage(std::vector <std::pair<Product, int>> products): A constructor that initializes the products member with the provided vector of product-quantity pairs.

- isAvailable(std::string name): Checks if a product with the given name is available in the storage.

- getProduct(std::string name): Retrieves the product with the given name from the storage.

- addProductToWaitingList(Product product): Adds a product to the waiting list in the storage.

- updateStorage(Product& product, int quantity\_sold): Updates the storage by reducing the quantity of the sold product and decreasing the waiting time for ordered products.

- getQuantityOnStorage(Product product): Retrieves the quantity of the given product in the storage.

* **Customer.h**

This header file defines the Customer class, which represents a customer of the pharmacy shop. It contains the following members:

- id: An ID representing the customer.

- name: A string representing the name of the customer.

- receipt: An integer representing the total cost of the customers purchases.

The Customer class provides the following member functions:

- Customer(ID id, std::string name): A constructor that initializes the id and name members with the provided arguments.

- getName(): Retrieves the name of the customer.

- getID(): Retrieves the ID of the customer.

- getReceipt(): Retrieves the total cost of the customers purchases

- addToReceipt(int newItem): Adds the cost of a new item to the customers receipt.

* **Drug.h**

This header file defines the base Drug class and its derived classes DrugNonPrescription and DrugWithPrescription. These classes represent different types of drugs available in the pharmacy shop. They contain the following members:

- name: A string representing the name of the drug.

- id: An ID representing the drug.

- price: A floating-point number representing the price of the drug.

- producer: A string representing the producer of the drug.

- waiting\_time: An integer representing the waiting time for the drug.

- need\_prescription: A boolean indicating whether the drug requires a prescription.

The Drug class is an abstract base class, and it provides a pure virtual function:

- printInformation() const: Prints information about the drug.

The derived classes, DrugNonPrescription and DrugWithPrescription, provide specific implementations for the printInformation() function.

* **Seller.h**

This header file defines the Seller class, which represents a seller in the pharmacy shop. It contains the following members:

- id: An ID representing the seller.

- name: A string representing the name of the seller.

The Seller class provides a constructor that initializes the id and name members with the provided arguments. The Seller class also has an overloaded << operator for printing the sellers information.