***Stock Market Simulator***

***SMSimulator Class***

* Класът SMSimulator е основният клас на симулационната програма.
* Той настройва GUI и координира различните компоненти на програмата.
* Класът SMSimulator отговаря за стартирането на симулацията и създаването на потребителския интерфейс.

***FormMediator class***

* FormMediator служи като посредник между различните форми в приложението: логин, регистрация и форма за забравена парола. Главната му задача е да управлява комуникацията и координацията между тези форми.

**Методи**

1. onLoginButtonClicked - Обработва събитието при клик на бутон за логин.

Проверява отчетните данни в базата данни и показва съответни съобщения.

1. onRegisterButtonClicked -

Обработва събитието при клик на бутон за регистрация.

Проверява дали паролите съвпадат и записва потребителя в базата данни, като показва съответни съобщения.

1. onForgotPasswordButtonClicked - Обработва събитието при клик на бутон за забравена парола.

Проверява дали паролите съвпадат и актуализира паролата на потребителя в базата данни, като показва съответни съобщения.

1. checkCredentials - Проверява отчетните данни на потребителя в базата данни.
2. saveUserToDatabase

* Записва потребителя в базата данни.

1. updateUserPassword

* Актуализира паролата на потребителя в базата данни.

***Mediator Class***

* Класът Mediator действа като посредник между различни компоненти в приложението Stock Market Simulator.
* Той получава заявки от компоненти и ги насочва към подходящия манипулатор.
* Той имплементира MediatorInterface, който дефинира методите за комуникация между компонентите.

***DatabaseHelper Class***

* Когато се инициализира чрез метода getInstance(), той създава единствена инстанция на себе си и настройва връзката към базата данни.
* Методът getConnection()позволява на други части от кода да получат връзка с базата данни.
* Всичко това помага за централизиране на връзката с базата данни и осигурява удобен достъп до нея в рамките на приложението.

***InvestorsCompaniesCreator Class***

* Методът "setCompanies" приема цяло число като параметър и генерира списък с компании с имена и стойности на приходите. Имената се вземат от масив от предварително дефинирани имена на компании.
* Методът "setInvestors" приема цяло число като параметър и генерира списък с инвеститори. Имената се вземат от масив от предварително дефинирани имена на инвеститори.
* Целта на този клас е да създаде списък с компании и инвеститори, които могат да се използват за симулатор на фондовия пазар.

***SimulationResultsSummary Class***

* Класът има един основен метод, наречен "getCompaniesCapital", който приема два параметъра ArrayList: "transactions" и "companies".
* Методът "getCompaniesCapital" връща ArrayList<Double>, съдържащ общия капитал за всяка компания в симулацията.
* Методът "getCompaniesCapital" изчислява общия капитал за всяка компания, като преминава през всяка транзакция и добавя цената на акциите на транзакцията към общия капитал на съответната компания.
* Този клас е ключов компонент при генерирането на обобщение на резултатите от симулацията, което може да се използва за оценка на представянето на всяка компания и инвеститор в симулацията.

***TradingDaySimulation Class***

* Класът TradingDaySimulation симулира търговията с акции между обекти на компания и инвеститор. Симулацията се извършва чрез метод tradingSimulator, който включва два обекта ArrayList: компании и инвеститори. Методът връща ArrayList от обекти Transaction.
* В рамките на цикъла методът произволно избира обект Company, който има акции за продажба, и обект Investor, който има достатъчно бюджет, за да закупи акциите. Създава се обект Transaction и се добавя към ArrayList на транзакциите. Цената на акциите на компанията се актуализира въз основа на броя продадени акции. Ако една компания продаде 10 акции, цената на акциите се удвоява. Ако се продадат 10 акции (от която и да е компания), а компанията не е продала нито една, цената се намалява с 2%.
* Методът “getCompaniesWithShares” връща ArrayList от обекти Company, които имат акции за продажба. Методът “getInvestorsWithEnoughBudget” връща ArrayList от обекти Investor, които имат бюджет, по-голям или равен на дадена сума.

***Company Class***

* Представлява компания в симулацията

***Investor Class***

* Играе ролята на инвеститор в симулацията

***GUIMenuBar Class***

* Представлява лента с менюта в графичен потребителски интерфейс.
* Той имплементира интерфейса Component и има обект MediatorInterface като членска променлива. Лентата с менюта има един елемент от менюто "Simulation", който съдържа елемент от менюто "New Simulation". Когато се щракне върху „Нова симулация“, ActionListener извиква метода newSimulation() на обекта медиатор.

***GUITableCompanies***

* Методът createTableCompanies приема ArrayList от компании като вход и преобразува данните в двуизмерен масив, който да се използва като данни в таблицата. Той задава имената на колоните като „ID“, „Name“, „Shares“ и „Share price“. Той създава JTable, използвайки имената на данните и колоните, и го обгръща в JScrollPane за функционалност за scroll-ване.

***GUITableInvestors***

* Класът GUITableInvestors е компонент от пакета за преглед на програмата SMSimulator.
* Той имплементира интерфейса Component и има обект MediatorInterface.
* Класът е отговорен за създаването на JTable за показване на данни на инвеститора в графичния потребителски интерфейс.

***GUITablesSummary***

* Той имплементира интерфейса Component и има обект MediatorInterface.
* Класът е отговорен за създаването на JTables за показване на обобщена информация за компании и инвеститори.
* Той генерира таблици с максимален/минимален капитал на компанията, максимален/минимален дял на инвеститорите и максимален/минимален брой компании на инвеститор.

***GUITableTarnsactions***

* Той имплементира интерфейса Component и има обект MediatorInterface.
* Този клас е отговорен за създаването на JTable за показване на информация за транзакция.
* Той генерира таблица с колони, представящи подробности за транзакцията, като ID, ID на инвеститора, ID на компанията, цена на акции на транзакция и дата.

***MainFrame***

* Той разширява класа JFrame и представлява основната рамка на приложението.
* Този клас отговаря за създаването и управлението на главния прозорец на приложението.
* Той предоставя методи за задаване на заглавие, размер, местоположение, икона и други визуални свойства на frame-а.

***LoginForm Class***

* Представлява форма за вход в приложението SMSimulator.

***RegistrationForm Клас***

* Представлява форма за регистрация в приложението SMSimulator.

***ForgotPasswordForm Клас***

* Представлява форма за забравена парола в приложението SMSimulator.