

**Технически университет – гр.Варна**

**Катедра СИТ**

**Курсов Проект**

По

Обектно-ориентирано програмиране - част 2

**Изготвили:**

Елена Райчева фак.н: 18621863

Калоян Маринов фак.н: 18621854 гр. Варна

група 1Б 2020г.

**V. Склад с наличности**

Да се разработи информационна система, предоставяща услуга склад. Програмата съхранява и обработва данни за складови помещения. Системата позволява множествен достъп.

Системата поддържа два вида потребители: администратор и оператори (складов агент) с различни роли за достъп до функционалностите в системата.

Операции за работа с потребители:

* Създаване на складови оператори от администратор;
* Създаване на доставчици;
* Създаване на клиенти;
* Създаване на каса (Парична наличност);

Система поддръжка операции за работа със събития:

* Създаване на номенклатури;
* Работа с фактури:
  + Приемане на стока от доставчик на доставна цена;
  + Изписване на стока на продажна цена;
* Наблюдение за наличност на стоки в склада;
* Наблюдение за наличност на пари в касата;

Системата поддържа справки по произволен период за:

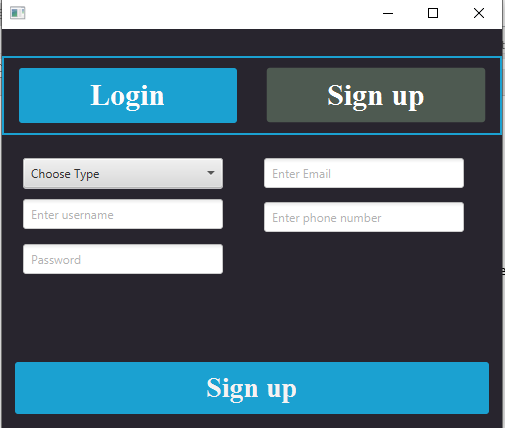
* Доставки и доставчици;
* Изписване и клиенти;
* Дейност на складовите оператори;
* За наличности в склада;
* Разходи, приходи, печалба;
* Движение на наличности в касата;

Системата поддържа Известия за събития:

* Критичен минимум и липса на стока;
* Критичен минимум и липса на парична наличност;

**Обяснения за работата на програмата**

1. Програмата се стартира от клас ***Login***, който зарежда прозорец, имащ две опции:



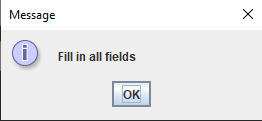
* **Създаване на Администратор (натискане на бутон “*Sign up*”) :**

След въвеждане на данни в полетата, при натискане на бутона „Sign up“

се активира функцията ***addAdministrator()***, намираща се в клас ***LoginController***.В нея се извършват следните действия:



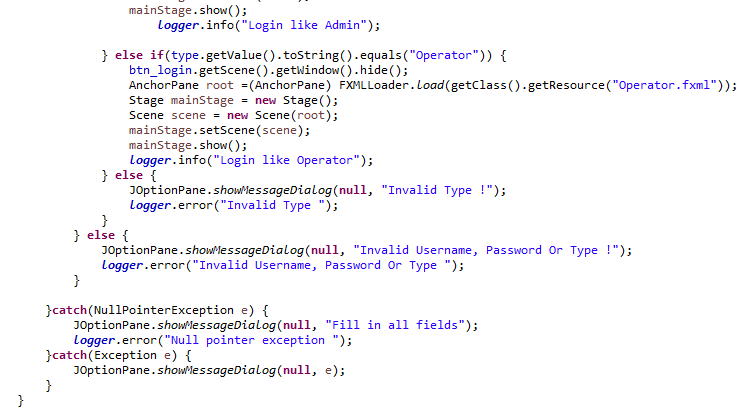
* Свързване с базата данни;
* Създаване на променливи, в които да се запазят данните от всяко едно от поле;
* Създаване на обект от класа ***Role***, който взима вече съществуващ запис в базата за Администратор;
* Създаване на обект от класа User, в който се поставят всички нужни стойности за създаване на нов ***User.*** Това са потребителско име, парола, имейл, телефонен номер и роля , която вече сме взели;
* След това, с функцията ***persist()*** се създава нов запис в базата данни в таблица Users с роля „***Administrator***“;
* Ако не всички полета са попълнени се зарежда прозорец:



* **Логване като Администратор или Оператор ( Натискане на бутон „*Login*” ) :**

След въвеждане на данни в полетата, при натискане на бутона „***Login***“ се активира функцията ***Login()***, намираща се в клас ***LoginController***.В нея се извършват следните действия:



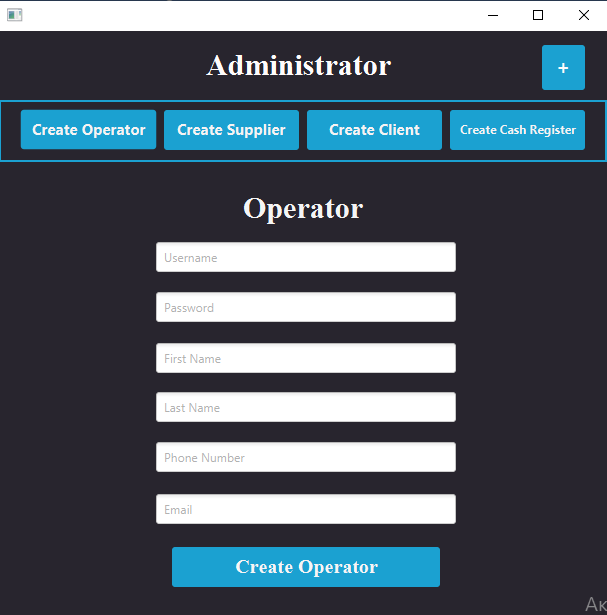


* Свързване с базата данни;
* Създаване на променливи, в които да се запазят данните от всяко едно от поле;
* В зависимост от избора се отварят различни прозорци за Оператор и Администратор;
* Потребителя, който е логнат се запазва в променливата ***userSave***;
* Ако не всички полета са попълнени се зарежда прозорец:

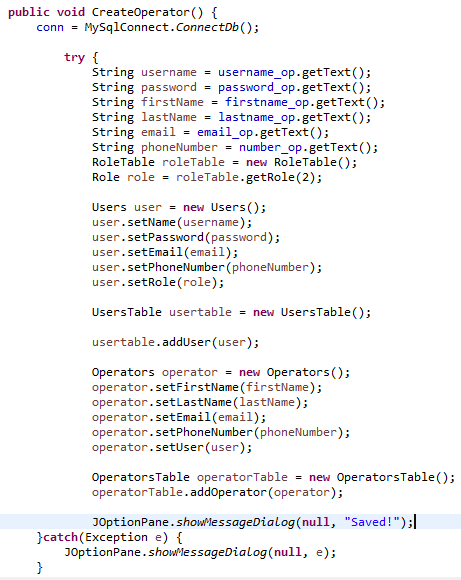


1. Възможности при влизане като Администратор :

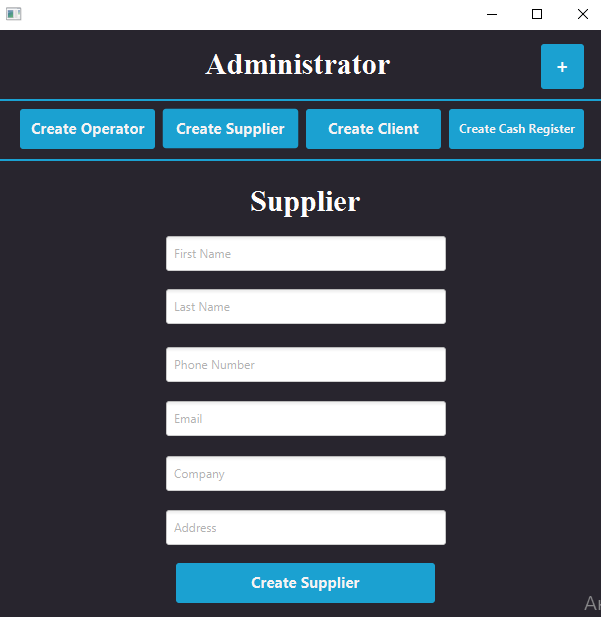
* **Създаване на оператор (натискане на бутон „*Create Operator*“) :**



След въвеждане на данни в полетата, при натискане на бутона „Create Operator“се активира функцията ***CreateOperator(),*** намираща се в клас ***AndministratorController***.В нея се извършват следните действия:



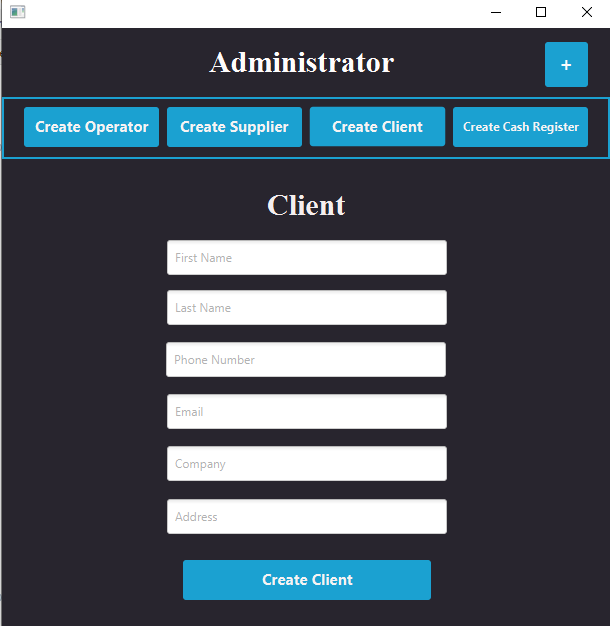
* Създаване на връзка с базата данни;
* Създаване на променливи, в които да се запазят данните от всяко едно от поле;
* Създаване на обект от клас ***Role***;
* Създаване на обект от клас ***Users***;
* Създаване на обект от клас ***Operato***r;
* Добавяне на създадения оператор в базата данни със съответния ***user***.
* **Създаване на доставчик (натискане на бутона „*Create Supplier*“) :**



След въвеждане на данни в полетата, при натискане на бутона „Create Supplier“се активира функцията ***CreateSupplier(),*** намираща се в клас ***AndministratorController***.В нея се извършват следните действия:



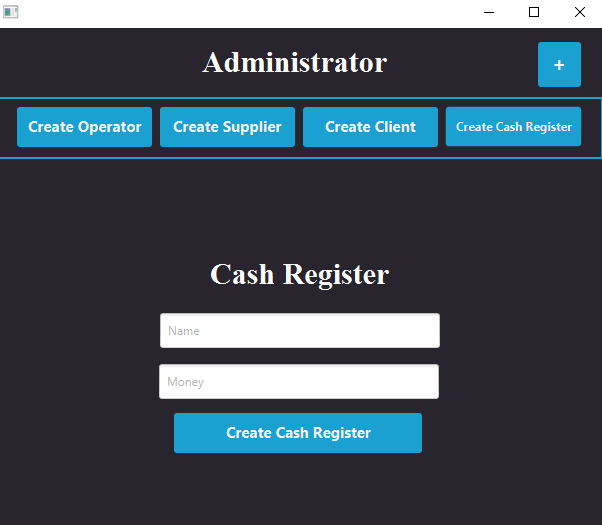
* Създаване на връзка с базата данни;
* Създаване на променливи, в които да се запазят данните от всяко едно от поле;
* Създаване на обект от клас Supplier;
* Добавяне на създадения доставчик в базата данни.
* **Създаване на клиент (натискане на бутона „*Create Client*“) :**



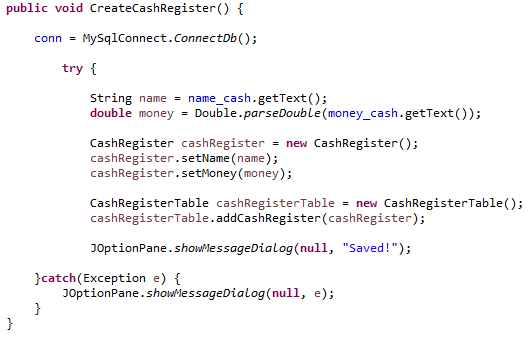
След въвеждане на данни в полетата, при натискане на бутона „***Create Client***“се активира функцията ***CreateClient()***, намираща се в клас ***AndministratorController***.В нея се извършват следните действия:



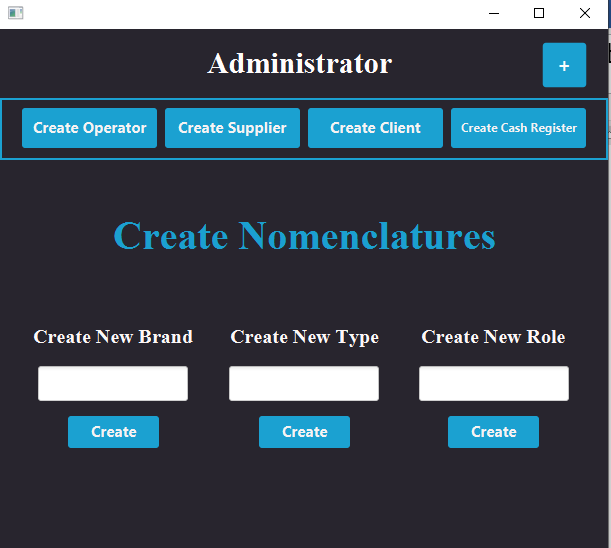
* Създаване на връзка с базата данни;
* Създаване на променливи, в които да се запазят данните от всяко едно от поле;
* Създаване на обект от клас Clients;
* Добавяне на създадения клиент в базата данни.
* **Създаване на каса (натискане на бутона „*Create Cash Register*“) :**



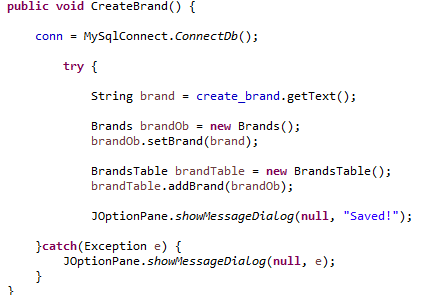
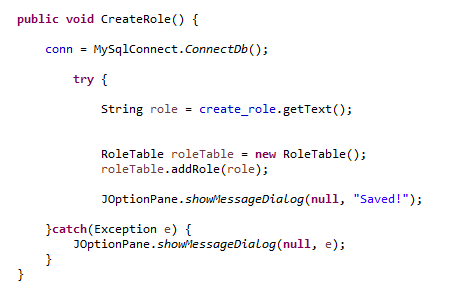
След въвеждане на данни в полетата, при натискане на бутона „***Create Cash Register***“се активира функцията ***CreateCashRegister()***, намираща се в клас ***AndministratorController***.В нея се извършват следните действия:

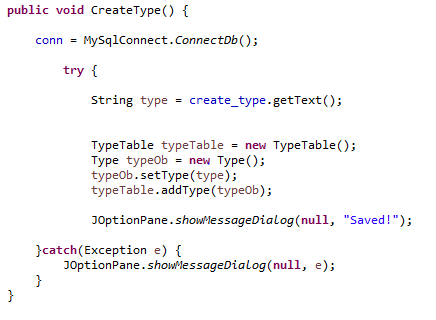


* Създаване на връзка с базата данни;
* Създаване на променливи, в които да се запазят данните от всяко едно от поле;
* Създаване на обект от клас ***CashRegister***;
* Добавяне на създадената каса в базата данни.
* **Създаване на номенклатури –марка, вид, роля(натискане на бутона „*+*“) :**



След въвеждане на данни в полетата, при натискане всеки от бутоните „Create“се активира съответната функция ***CreateRole(), CreateType()*** или ***CreateBrand()***, намиращи се в клас ***AndministratorController***. В тях се извършват следните действия:



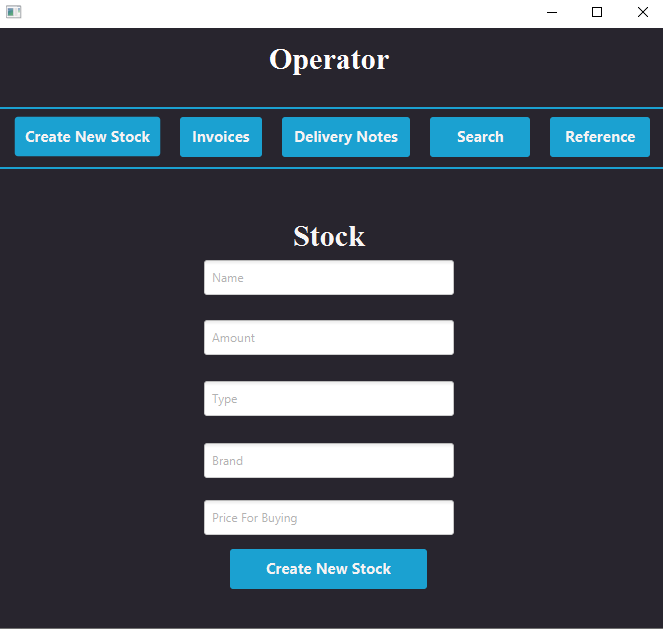


Всяка една функция:

* създава връзка с базата данни;
* създава променлива, в която се запазва названието съответно на марката, вида и ролята;
* добавя стойността на променливата в базата данни.

1. Възможности при влизане като Оператор :

* **Създаване на нова стока (натискане на бутона „*Create New Stock*“):**

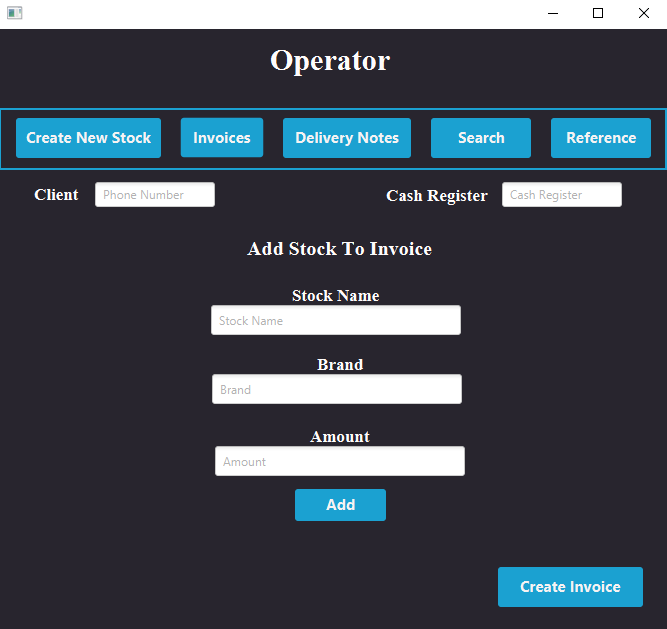


След въвеждане на данни в полетата, при натискане на бутона „***Create New Stock***“се активира функцията ***CreateStock(),*** намираща се в клас ***OperatorController***.В нея се извършват следните действия:

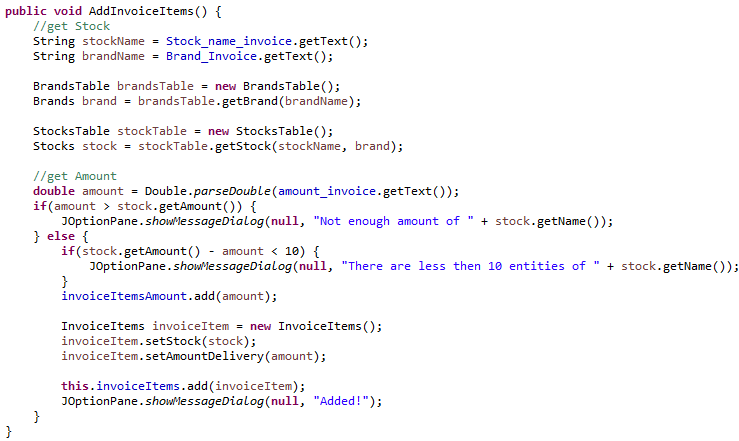




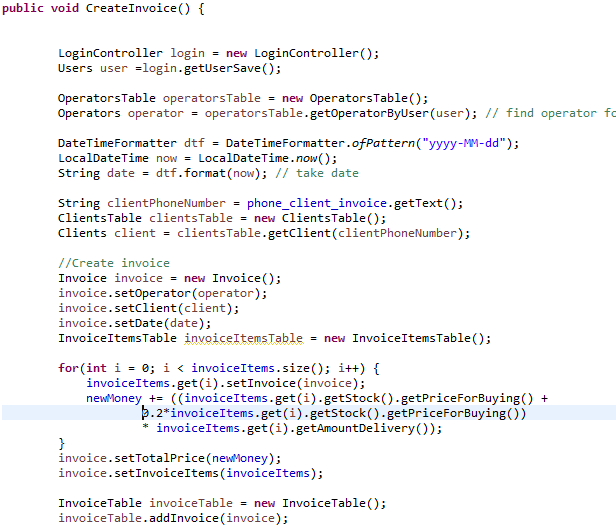
* Създаване на връзка с базата данни;
* Създаване на променливи, в които да се запазят данните от всяко едно от поле;
* Създаване на обект от клас ***Brands*** на съответната стока;
* Създаване на обект от клас ***Stock***;
* Добавяне на ролята в базата данни;
* **Създаване на фактура(изписване на стока) (натискане на бутона „*Invoices*“) :**

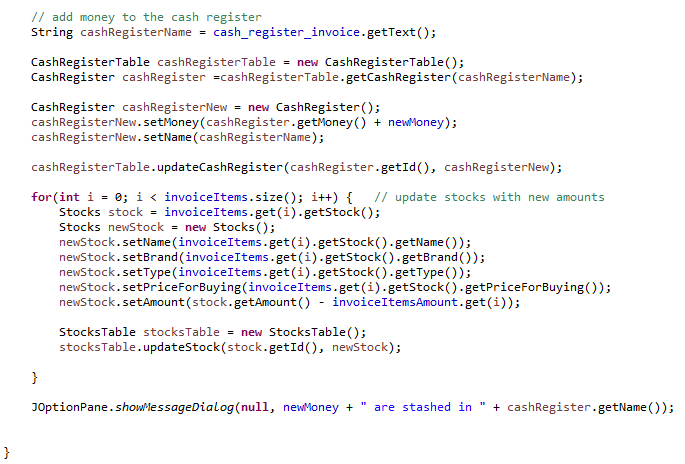


Изписването на стока се изпълнява чрез две функции: Първо, чрез попълването на полетата “***Stock Name***”, “***Brand***”, ”***Amount***” и натискането на бутона „***Add***“ се добавят отделните стоки и количествата им (функция ***AddInvoiceItems()***). Когато сме въвели всички стоки на фактурата, се попълва текстово поле на “***Client***“. Клиентът, на когото се издава фактурата, се търси в базата данни чрез телефонен номер. Попълва се и полето на “***Cash Register***”. Касата, в която ще се генерират парите от фактурата, се търси в базата данни по име. След това фактурата се създава, чрез натискане на бутона „***Crеаtе Invoice***“ (функция ***CreateInvoice()***).

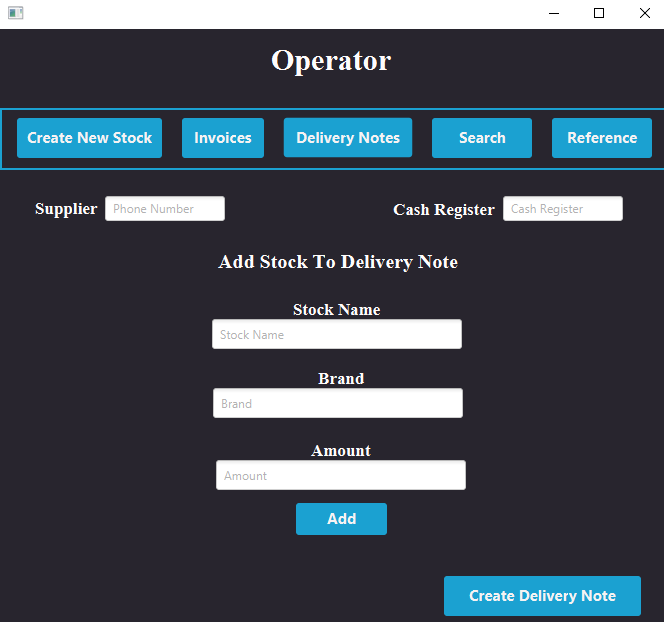


* Създаване на променливи, в които да се запазят данните от полетата;
* Намиране на бранда в базата данни;
* Намиране на стоката в базата данни;
* Проверка дали има достатъчна наличност в склада от съответната стока
* Създаване на обект от клас ***InvoiceItems***;
* Добавяне на обекта към ***List invoiceItems***.

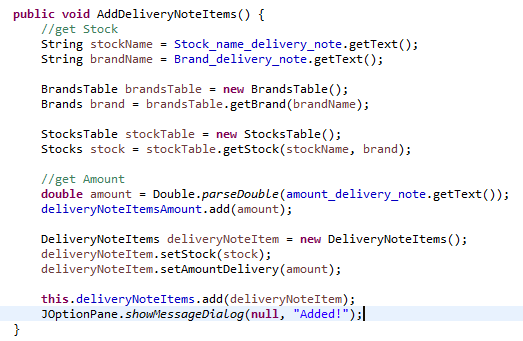




* Създаване на обект от клас ***Users***, в който се запазва user-а, който е log-нат.
* Намиране на оператора, чрез user-а;
* Взимане на дата;
* Намиране на клиента, от телефонния му номер;
* Създаване на обект от клас Invoice.
* Добавяне на всички обекти от листа ***invoiceItems*** към фактурата и добавяне на общата им стойност + 20% печалба (брой \* единична цена + 0,2 \* брой \* единична цена );
* Намиране на касата по име и добавяне на парите;
* Промяна на количества в склада на всяка една от продадените стоки;
* **Създаване на фактура(приемане на стока) (натискане на бутона „*Delivery Notes*“) :**

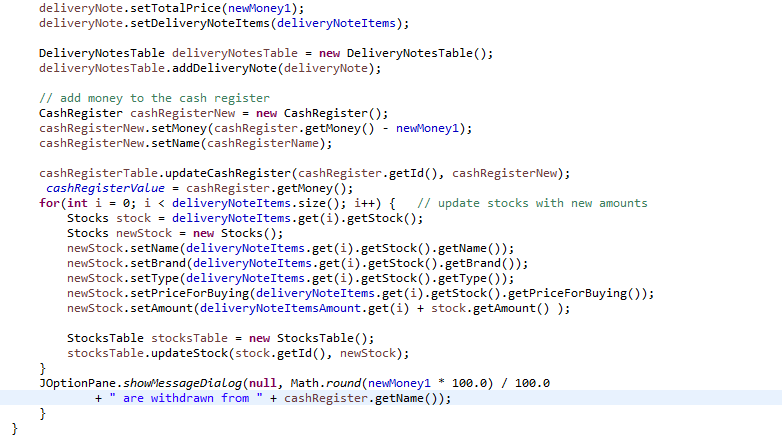


Приемането на стока се изпълнява чрез две функции: Първо, чрез попълването на полетата “***Stock Name***”, “***Brand***”, ”***Amount***” и натискането на бутона „***Add***“ се добавят отделните стоки и количествата им (функция ***AddDeliveryNoteItems()***). Когато сме въвели всички стоки на фактурата, се попълва текстово поле на “***Supplier***“. Доставчикът се търси в базата данни чрез телефонен номер. Попълва се и полето на “***Cash Register***”. Касата, от която ще се вземат парите от фактурата, се търси в базата данни по име. След това фактурата се създава, чрез натискане на бутона „***Crеаtе Delivery Note***“ (функция ***CreateDeliveryNote()***).

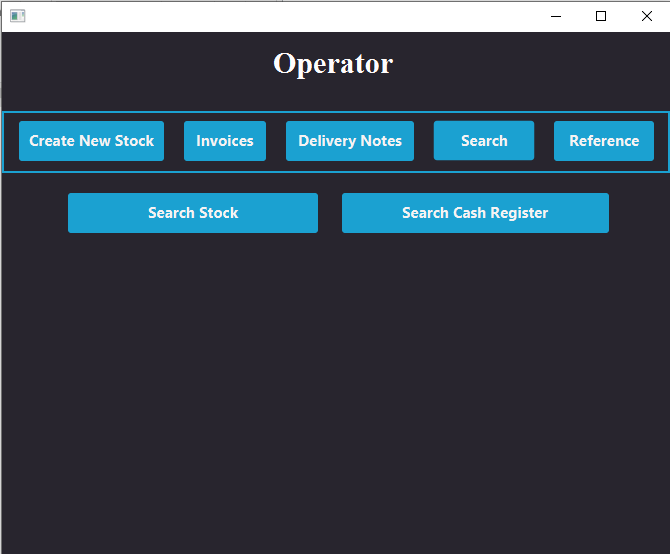


* Създаване на променливи, в които да се запазят данните от полетата;
* Намиране на бранда в базата данни;
* Намиране на стоката в базата данни;
* Създаване на обект от клас ***deliveryNoteItems***;
* Добавяне на обекта към ***List deliveryNoteItems***.

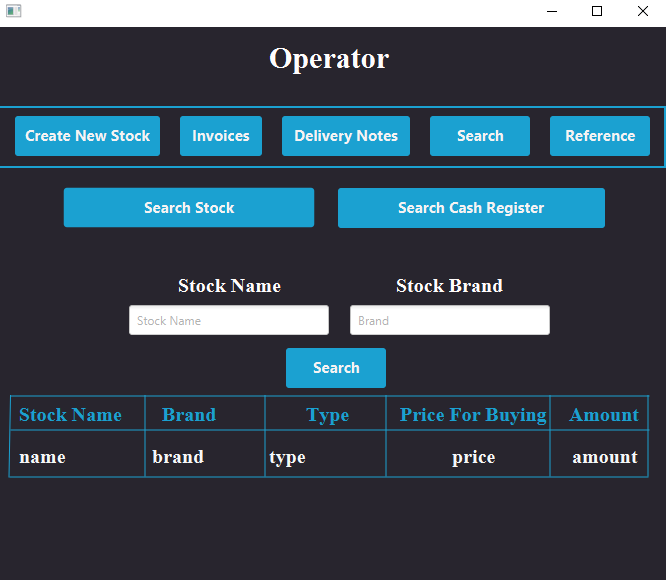




* Създаване на обект от клас ***Users***, в който се запазва user-а, който е log-нат.
* Намиране на оператора, чрез user-а;
* Взимане на дата;
* Намиране на доставчика, от телефонния му номер;
* Създаване на обект от клас ***DeliveryNotes***.
* Проверка за наличие на пари в касата
* Добавяне на всички обекти от листа ***deliveryNoteItems*** към фактурата и добавяне на общата им стойност (брой \* единична цена );
* Намиране на касата по име и изваждане на парите;
* Промяна на количества в склада на всяка една от продадените стоки;
* **Търсене (натискане на бутона „*Search*“):**



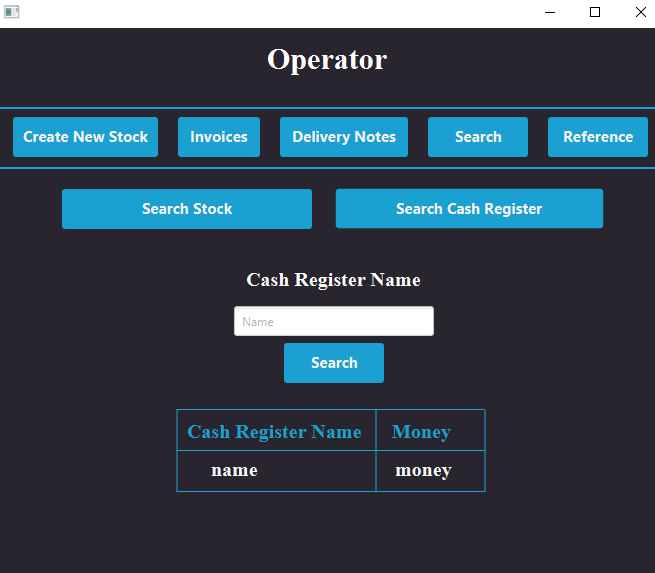
* **Търсене на стока (натискане на бутона „*Search Stock*“)**



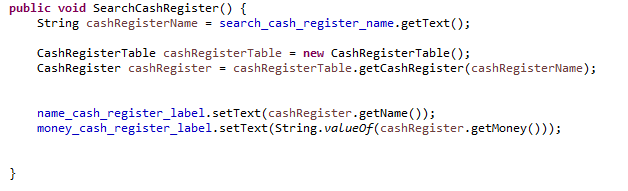
След въвеждане на данни в полетата(***Stock Name*** и ***Stock Brand***), при натискане на бутона „***Search***“се активира функцията ***SearchStock(),*** намираща се в клас ***OperatorController***. В нея се извършват следните действия:



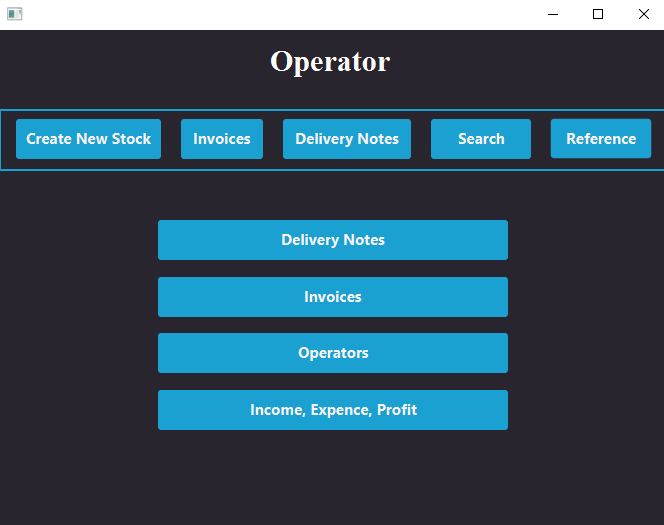
* Създаване на променливи за запазване на име и марка;
* Намиране на стоката в базата данни по запазените име и марка;
* Промяна на label полетата със стойностите на намерената стока;
* **Търсене на каса (натискане на бутона „*Search Cash Register*“)**



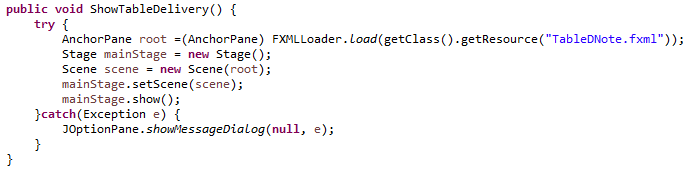
След въвеждане на име на каса ***Cash Register Name***, при натискане на бутона „***Search***“се активира функцията ***SearchCashRegister()***, намираща се в клас ***OperatorController***.В нея се извършват следните действия:



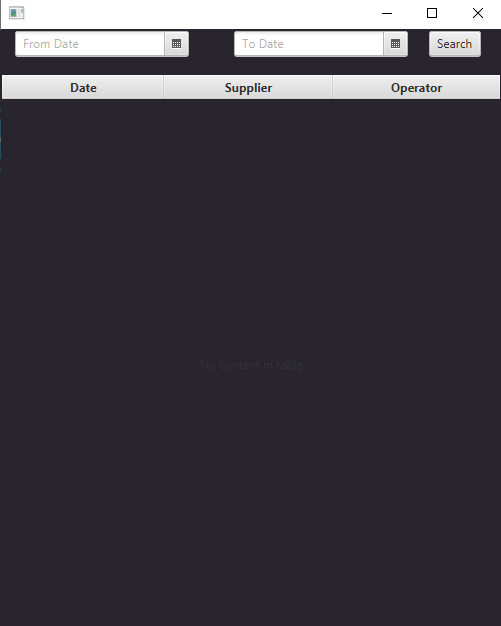
* Създаване на променлива за запазване на име на каса;
* Намиране на касата в базата данни по име;
* Промяна на label полетата със стойностите на намерената каса;
* **Справки по произволен период (натискане на бутона „*Reference*“) :**



* **Справки за доставки и доставчици (натискане на бутона *Delivery Notes*):**

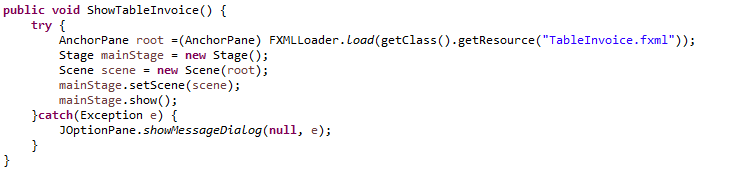


При натискане на бутона „***Delivery Notes***“се активира функцията ***ShowTableDelivery()***, намираща се в клас ***OperatorController***. Чрез нея се зарежда нов прозорец, за който отговаря клас ***TableDeliveryController***.

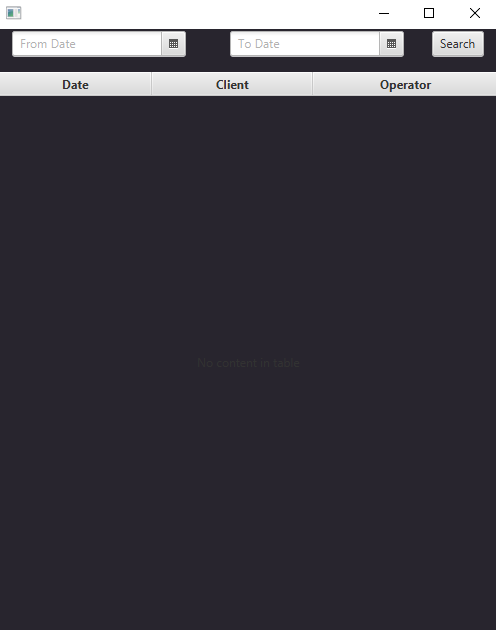




* Създаване на връзка с базата данни;
* Запазване на датите, избрани от потребителя, в променливи;
* Избиране от базата данни на фактурите (за приемане на стока) от зададения период;
* Цикъл за запазване на резултата в листа от тип ***DeliveryNoteModel*** (създаден за представяне на информацията в таблицата);
* Представяне на резултата от заявката в таблицата;
* **Справки за изписване и клиенти (натискане на бутона „*Invoices*“) :**



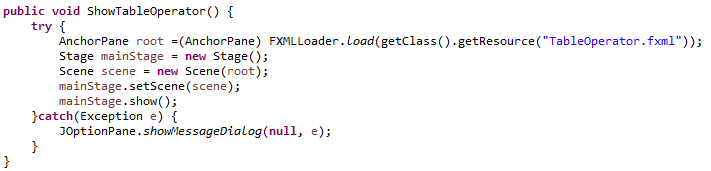
При натискане на бутона „***Invoices***“се активира функцията ***ShowTableInvoices()***, намираща се в клас ***OperatorController***. Чрез нея се зарежда нов прозорец, за който отговаря клас ***TableInvoiceController***.



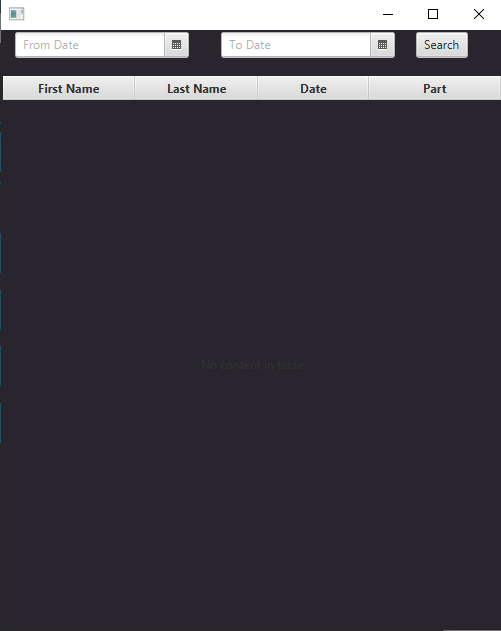
****

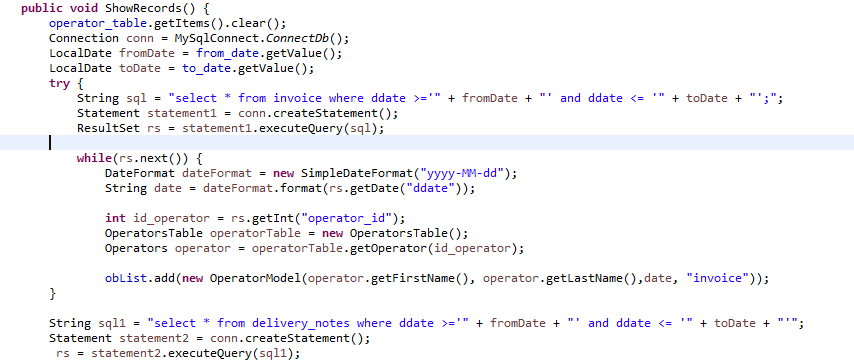
* Създаване на връзка с базата данни;
* Запазване на датите, избрани от потребителя, в променливи;
* Избиране от базата данни на фактурите (за издаване на стока) от зададения период;
* Цикъл за запазване на резултата в листа от тип ***InvoiceModel*** (създаден за представяне на информацията в таблицата);
* Представяне на резултата от заявката в таблицата;

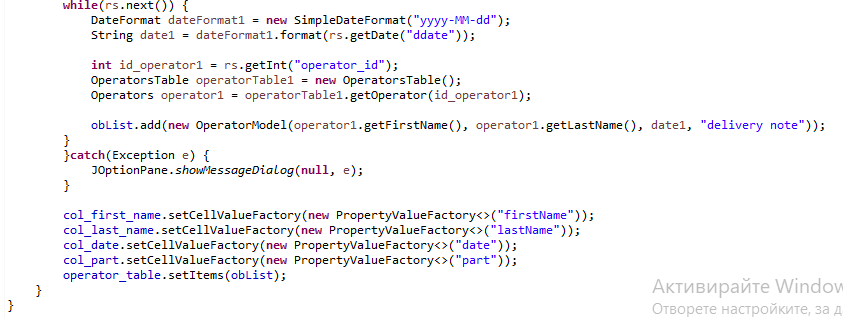
* **Справки за дейности на складовите оператори (натискане на бутона „*Operators*“) :**



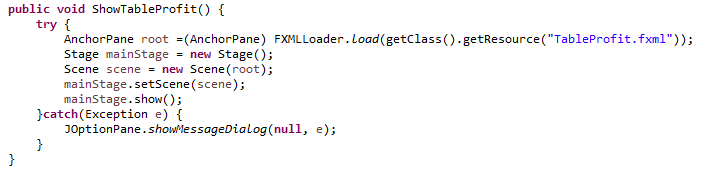
При натискане на бутона „***Operators***“се активира функцията ***ShowTableOperator(),*** намираща се в клас ***OperatorController***. Чрез нея се зарежда нов прозорец, за който отговаря клас ***TableOperatorController***.



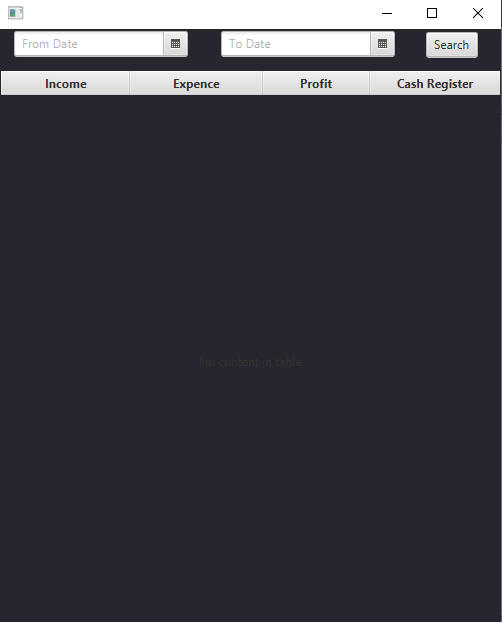




* Създаване на връзка с базата данни;
* Запазване на датите, избрани от потребителя, в променливи;
* Избиране от базата данни на фактурите (за издаване на стока) от зададения период;
* Цикъл за запазване на резултата в листа от тип ***OperatorModel*** (създаден за представяне на информацията в таблицата);
* Избиране от базата данни на фактурите (за приемане на стока) от зададения период;
* Цикъл за запазване на резултата в листа от тип ***OperatorModel*** (създаден за представяне на информацията в таблицата);
* Представяне на резултата от заявката в таблицата;
* **Справки за разходи, приходи, печалба (натискане на бутона „*Income, Expence, Profit*“) :**



При натискане на бутона „***Income, Expence, Profit***“се активира функцията ***ShowTableProfit()***, намираща се в клас ***OperatorController***. Чрез нея се зарежда нов прозорец, за който отговаря клас ***TableProfitController***.





* Създаване на връзка с базата данни;
* Запазване на датите, избрани от потребителя, в променливи;
* Избиране от базата данни на всички каси;
* Избиране от базата данни на фактурите (за приемане на стока) от зададения период и запазване на общата стойност(разход);
* Избиране от базата данни на фактурите (за изписване на стока) от зададения период и запазване на общата стойност(приход);
* Запазване на разликата между прихода и разхода като печалба
* Запазване на резултата в листа от тип ***ProfitModel*** (създаден за представяне на информацията в таблицата);
* Представяне на резултата от заявката в таблицата;

1. Към всеки от основните класове има клас, който работи с базата данни

* Brands – BrandsTable
* Type – TypeTable
* Stocks – StocksTable
* Role – RoleTable
* Users – UsersTable
* Operators – OperatorsTable
* Supplier – SupplierTable
* Clients – ClientsTable
* CashRegister – CashRegisterTable
* Invoice – InvoiceTable
* InvoiceItems – InvoiceItemsTable
* DeliveryNotes – DeliveryNotesTable
* DeliveryNotesItems – DeliveryNotesItemsTable