**MINISTRY OF EDUCATION AND SCIENCE OF KYRGYZ REPUBLIC**

**KYRGYZ-GERMAN INSTITUTE OF APPLIED INFORMATICS**

**Course Project:**

**Subject: «** **Automation for the selection of materials to repair**

**apartments»**

**Teacher: Almazbek u. A.**

**Students: Group №3 (AIN-2-20)**

**Bakaev Dastan**

**Kushtarbekov Kubanychbek**

**Kubatbekov Alaken**

**Sharshenkulov Aidar**

**Ismailov Ersultan**

**Bishkek 2021**

Contents

**Introduction: ………………………………….**

**Team Roles and Responsibilities: ……….**

**Description and App Architecture: ……………………………**

**Сode review: ……………………………………….**

**Login system – Sharshenkulov Aidar ……………………….**

**Owner account– Kushtarbekov Kubanychbek ………...**

**Delivery account – Ismailov Ersultan**

**Bakaev Dastan …………………….**

**Foreman account –Kubatbekov Alaken ……………………**

**Conclusion: ……………………………………...**

**Literature: …………………………………**

# Introduction

The work is carried out to facilitate the accounting and selection of materials required for the repair of apartment premises. Several people monitor the availability of materials and they need this kind of program. Among these people are: Foreman, Owner, Delivery man.

1. Purpose of the development

"Automation of the selection of materials for apartment repairs" - is designed to summarize information about the availability of certain building materials that may be necessary for apartment repairs, depending on the wishes and needs of a particular client.

Users of the program are the Foreman, the Owner of the apartment. Everyone will have their own access to the system by login and password.

* For a particular host, for example, login: "owner1” and password

"pa$$w0rd123"(Or you can set it at your own discretion)

* For the caretaker, for example, login: "prorab" and password "zavkhoz123" (Or you can, at your discretion, you can set)
* For the deliverer, for example, the username "delivery" and the password "delivery123" (Or you can set it at your discretion)

The username and passwords can be read from the file and checked in conditional

Constructs. Logins and passwords for the "Foreman" account are stored in a file

"password.txt", and the passwords for the nth number of apartment owners are stored in a separate file "owners.txt". Depending on the account category of a particular person, the corresponding menu opens. All the names of building materials with the names of companies and prices must be present in a separate file “materials-price.txt” General information on the quantity of construction materials, completion the name of the object, the name of all building materials is available to builders. Information about the quantity and price of building materials for your object is visible to the owner of the apartment.

The deliverer has the ability to see how much construction materials were delivered, and depending on the amount of construction materials, the program automatically calculates the price for delivery.

The Foreman can also remove, add, or change the price for building materials. Can also add to the list of hosts in the file "owners.txt»

2. Program Requirements

2.1. Requirements for functional characteristics The automated information system "Educational Center" must provide the following functions:

* Login to the account
* The ability to choose from the menu based on the type of account of the person
* Ability to search for building materials, change the price
* Ability to see the price, purchase date, and more

2.2. Program performance scenario

Host Account

1. The program starts and offers the first thing to do is to keep the account type and

type the keyword.

Example:

To start the program, please enter the account type: >>> owner1

If the name of the account type does not match, the program ends with the words:

Sorry, but we did not find this type of account, please repeat.

2. Then you are prompted to enter your username and password, and after successful authorization, the menu for the host opens.

A separate file is created for each host “owner1.txt”, “owner2.txt” etc, where these files contain lists of building materials, the name, the manufacturer and the prices for each of the building materials.

An example menu for a caretaker looks like this:

Greetings dear, Host 1 (or host Adilet)!

Please dial the menu number to work with the program, if you are finished, then dial 6:

1. Show the entire list of building materials for my apartment

2. Search for building materials:

• By the name of the building materials

\* Write the name of the building material to display:>>

(After which a list of the building material that meets the specified search criteria is displayed)

• According to the manufacturer's company

\* Write the name of the manufacturer to search for:>> (After which a list of materials for this manufacturer is displayed)

3. Order construction material

\* Write the name of the building material to order:>>

(After the user writes the name of the building material that he wants to order in a file named "ordered materials.txt" the material for the order is added with the name or account of the owner)

4. Withdraw the order

\* Write the name of the building material in order to revoke:>> (After the user writes the name of the building material that he wants to delete or revoke, from the file named "ordered-materials.txt" this or that material that the owner would like to recall is removed)

5. View the list of ordered building materials (Shows the entire list of ordered building materials from the "Ordered Building Materials" file»)

6. Exit

Menu selection: >>

If the "Exit" option is selected, the program ends with the words: The program is completed; we look forward to your return!

Delivery Account

1. The program starts and offers the first thing to do is to maintain the account type:

Example:

To launch the program, please enter your account type and your keyword: >>> Deliveryman

If the name of the account type does not match, the program it ends with the words:

Sorry, but we did not find this type of account, please repeat.

2. Then you are prompted to enter your username and password, and after successful authorization, the menu for the deliverer opens.

An example menu for a deliverer looks like this:

Greetings, dear Delivery Guy!

Please dial the menu number to work with the program, if you are finished, then dial 7:

1. Show the list of materials for delivery

(Shows the list of materials to be delivered from the file "ordered\_materials.db”)

2. Show the delivered materials

(Shows the list of delivered hardware from the file "delivered\_materials.db")

3. Deliver building material:

• What building material was delivered?

Enter the name of the equipment or its serial number >>>

(To database "delivered\_materials.db" enter the name, the company of the building material and the name of the owner for whom it was delivered by the deliverer and simultaneously delete the record of this material from the database "ordered\_materials.db»)

4. Show the number of delivered building materials:

(Shows the number of delivered building materials from database "delivered\_materials.db»)

5. Show the quantity of ordered materials building

(Shows the number of construction materials to be delivered from "ordered - materials.txt»)

6. Show my earnings

(Shows how much the deliverer will receive for each delivered building material from the file "delivered-items.txt", the delivery rate can be set for your consideration)

7. Exit (Exits the program) Menu selection: >>>

If the "Exit" option is selected, the program ends with the words: The program is completed; we look forward to your return!

Foreman's Account

1. The program starts and offers the first thing to do is to keep the account type and type the keyword.

Example:

To start the program, please enter the account type: >>> Prorab

If the name of the account type does not match, the program ends with the words:

Sorry, but we did not find this type of account, please repeat.

3. Then you are prompted to enter your username and password, and after successful authorization, the menu for the foreman opens. An example menu for a foreman looks like this:

Greetings, dear Foreman!

Please dial the menu number to work with the program, if finished, then dial 7:

1. Show a list of all apartment owners

(Shows a list of all hosts with their data from the file "owners.txt»)

2. Show a list of all building materials

\* Write the name of the owner of the apartment, the materials of which you want to display >>> After entering the name of the apartment owner, all its materials are displayed, example: owner1 -> all building materials for this apartment owner are displayed)

3. Show the building material with the highest price

\* Type the name of the owner>> (Shows the building material with the highest price for a specific owner)

4. Show the building material with the lowest price

\* Type the name of the owner>>

(Shows the building material with the highest price for a particular landlord)

5. Show the average price of building materials

(Shows the average price of building materials from the file "material-price.txt")

6. Perform the Delete-Add-Edit operation

\* Select the option:

* Remove building material
* Change the building material
* Add building material

(Depending on the user's choice of one of the options you can delete, add, or change the building material in the file "material-price.txt»)

7. Exit (Exits the program) Menu selection: >>>

If the "Exit" option is selected, the program ends with the words: The program is completed, we look forward to your return!

The program must be written using the Java language and all write and delete operations must be performed by writing to a file, reading from a file, and performing operations on them.

# Team Roles and Responsibilities:

As a semester project for the class “Programming Languages (Java)” we have been given a task to complete in a group of five.

**Project name:**

“ Automation of the selection of materials for apartment repairs ”

**Objective:**

Implement a database navigation program to summarize information for certain building materials that may be necessary for the repair of an apartment, depending on the wishes and needs of a particular client.

**Technologies used:**

* Java
* Java libraries:
* Scanner
* io
* Command line

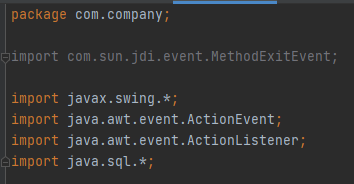
**Team Roles and Responsibilities (Alphabetical Order):**

1. **Sharshenkulov Aidar** – *backend developer*
2. **Ersultan Ismailov** – tester, documentation
3. **Kushtarbekov Kubanychbek** – backend developer
4. **Kubatbekov Alaken –** backend developer
5. **Bakaev Dastan** – backend developer

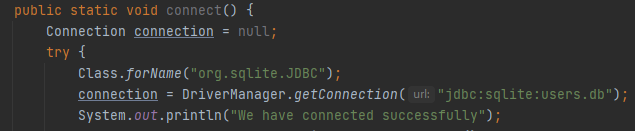
# Code review

# Login

First of all, we are import all the libraries that we want to use



Since we are using SQLite in our course project, we must connect our IntellijIdea with SQLite through the code.

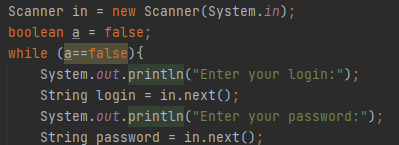


After we have connected to our database we can use functions that are using is SQL.

We used the function that open our database and execute the code that we are enter to our SQLite.



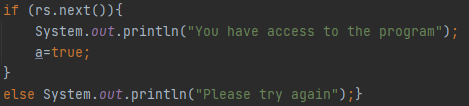
Then we open endless while loop that connect with the Boolean variable and in this while loop we use Scanner to scan the values that users will input. Firstly, user must print his login and then the password. System.out.println() notice user where they should enter their login and password.



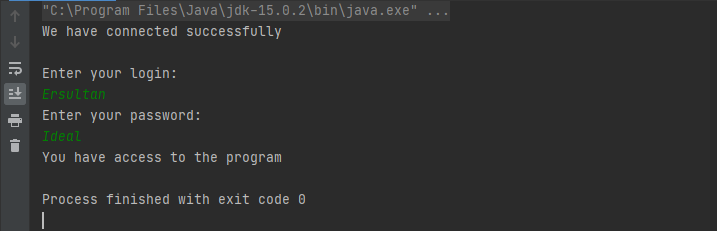
After the user have entered their login and password we are using the function that execute this in SQLite and give the result to ResultStatement.



Then we are using if condition to check if the login and password are in database. If the condition pass we are giving the access to user through Boolean variable. And if the user print incorrect answer, he can enter it one more time



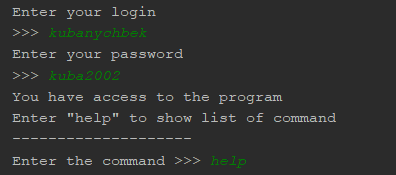
Running the program



# Account of owner:

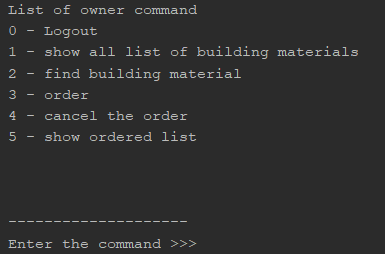
After entering to account of Owner:

There will be appeared a menu of methods:

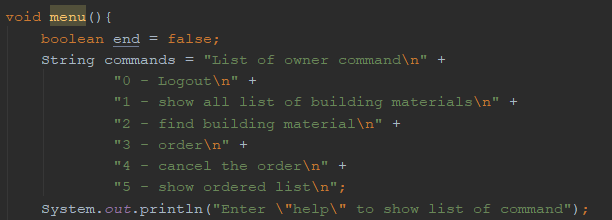


To show list of commands, user(owner) enters “help”

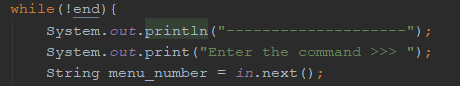
And the list of commands will be appeared:



Code of menu:

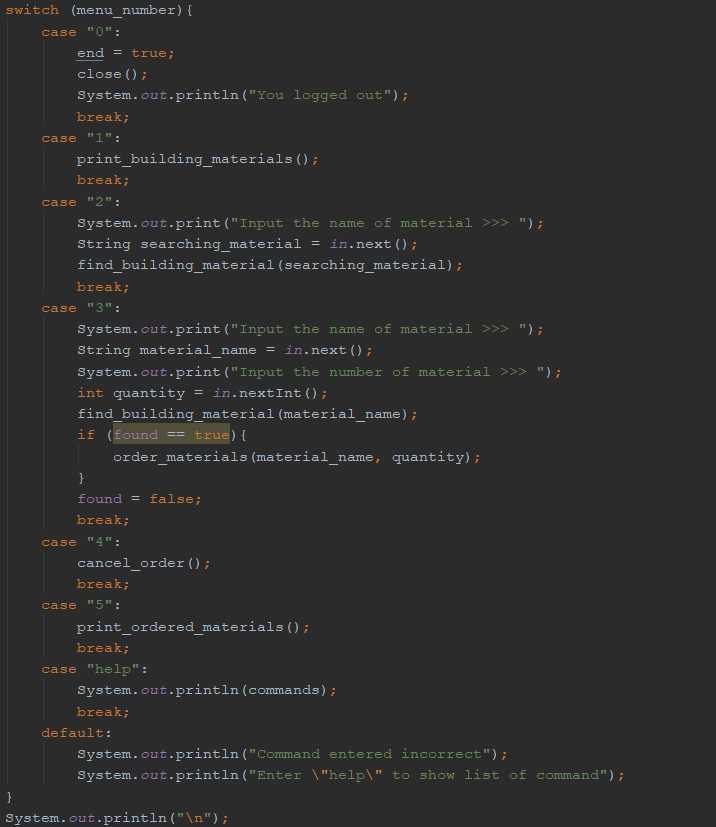


Here we created variables: commands(String) – this variable contains list of commands that user can use, end(boolean) – this variable is used for our endless loop(while).



In this loop user enters certain command and this command will be executed

The variable saves the command that enters user



And through switch the program checks what users enters.

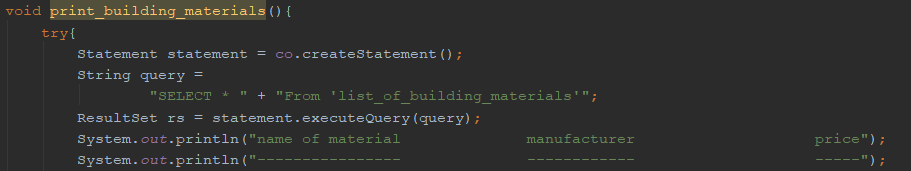
There are some cases:

1. If user enters “1”, will be executed method “print\_building\_materials”;
2. If user enters “2”, will be executed method “find\_building\_material”;
3. If user enters “3”, will be executed method “order\_materials”;
4. If user enters “4”, will be executed method “cancel\_order”;
5. If user enters “5”, will be executed method “print\_ordered\_materials”;
6. If user enters commands out of our list of commands, will be appeared message “Command entered incorrect” and will require to enter “help”:
   * If user enters “help”, will be showed list of commands.
7. If user enters “0”, user will logout the account of owner.

Let’s consider how methods(commands) of the owner’s account work

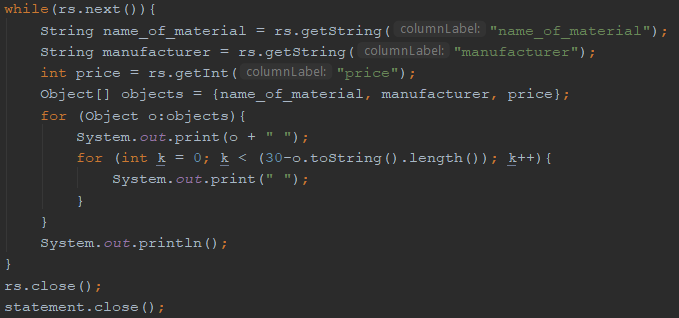
1. print\_building\_materials()

Code of this program:



Before printing our list of building materials, here we first create statement in order to execute our simple SQLite queries. Then there is the variable “query” that contains query to our statement. And variable rs(ResultSet) gets results from statement.executeQuery(query).

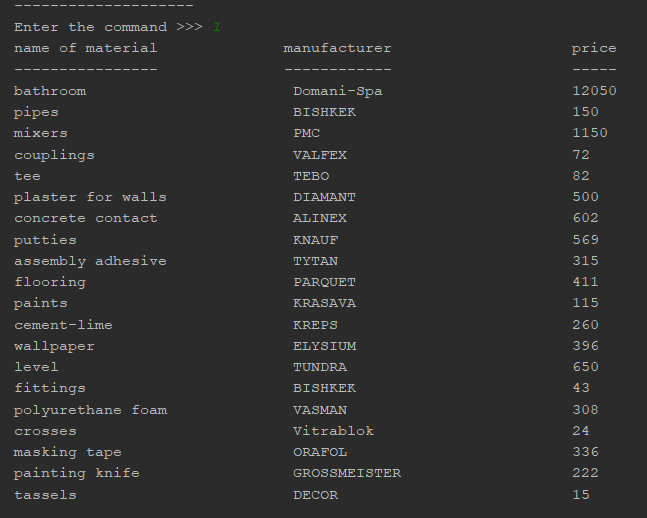
Then there some messages: the names of columns of our database



And here the program will print values through loop(while). There is also another loop(for) of our primary loop, with helping of this loop(secondary), our values will be printed in straight.

Like here:

User enters “1” and list of building materials will be printed

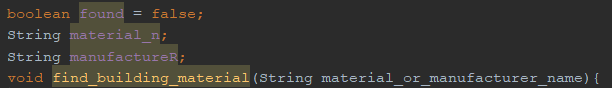


1. Find a building material:

Code:

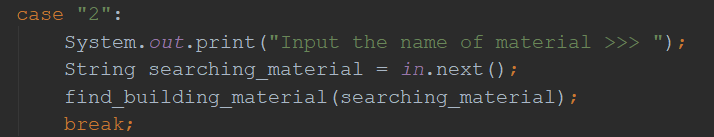
The code of this method is the same as previous method “print\_building\_materials”.

There is a few differents:

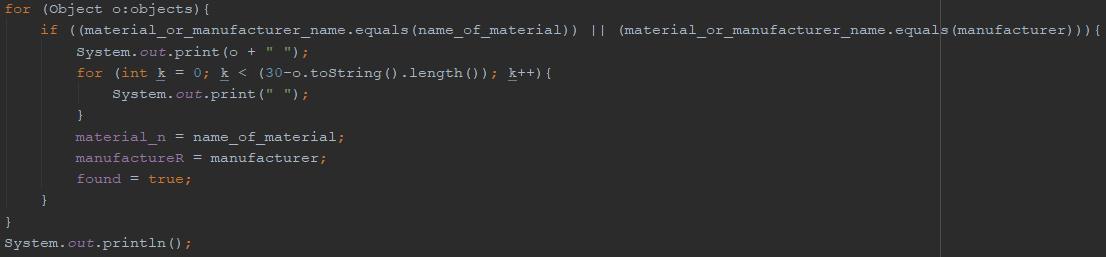


These variables are used for saving values if user finds the materials that looks for.

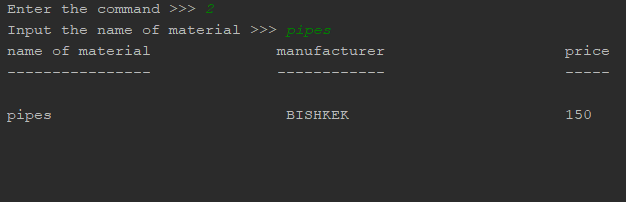
Before this method takes a parameter “material\_or\_manufacturer\_name”. This parameter user indicates in menu:



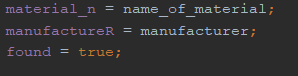
There is a condition in the loop, here program compares “material\_or\_manufacturer\_name” with “name\_of\_material” or with “manufacturer”



If user enters a material that is contained in list of materials, the console will print this material:

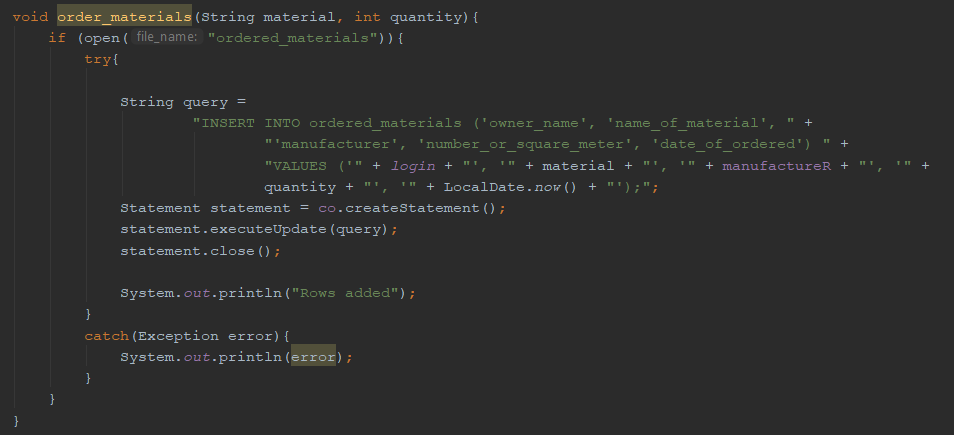


After finding the material the variables are updated:



1. Ordering building materials

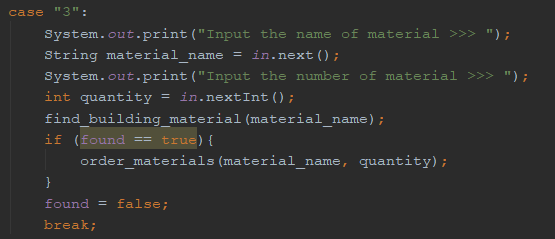
Code:



To order materials, we first open another database called “ordered\_materials”

Then there is created statement in order to execute our simple SQLite queries. Then there is the variable “query” that contains query to our statement. And variable rs(ResultSet) gets results from statement.executeQuery(query).

Also this method takes two parameters. They are indicated in menu:



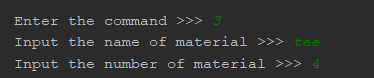
User enters values: material name and number of material.

And method takes these values.

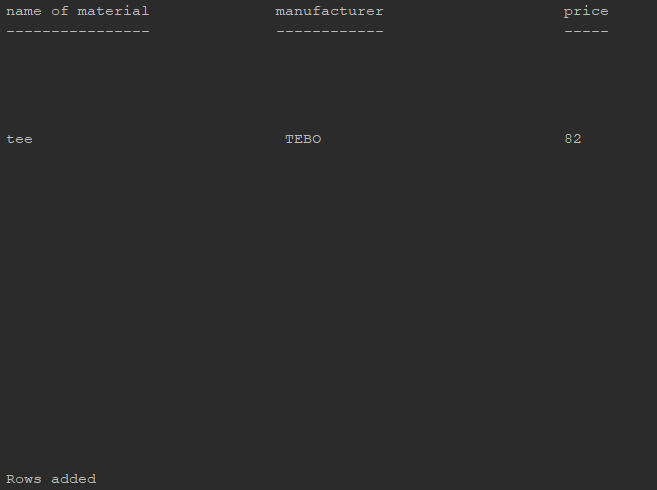
After as you can see that the variable “found” takes value “false”. It’s used for not getting bug.

Bug: if found is not false, then the program orders materials that is not our list of materials.

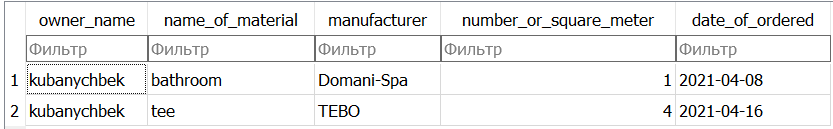
To order building materials, user enters a name of this material and number:



If the “name of material” is contained in list of building materials, the program adds new record in database and prints information about this materials and prints message “rows added”:

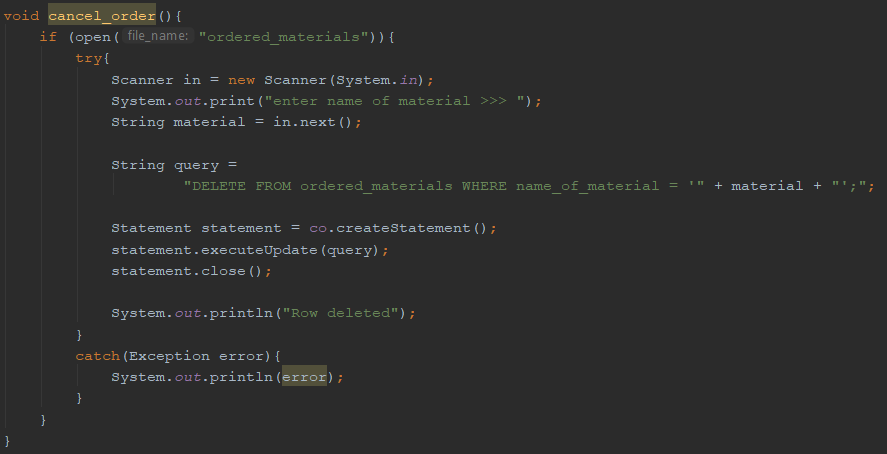


This information is saved to file “ordered\_materials.db”:



1. Cancel the order:

Code:

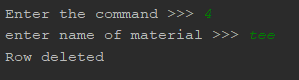


This method is also same as method “order\_materials”. There is changed the value in the variable query. This query deletes a row.

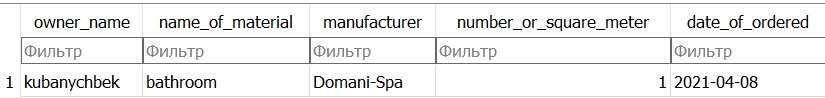
To cancel the order, user enters the material name. And if database’s rows contain the material name entering by user, this rows will be deleted.

If the order is cancelled, there will be appeared the message “Row deleted”

To cancel the order, user enters a name of this material:

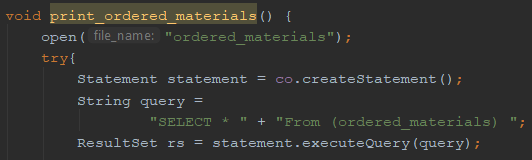


And then the information will be deleted from file “ordered\_materials.db”:



1. Print ordered construction materials:

Code:



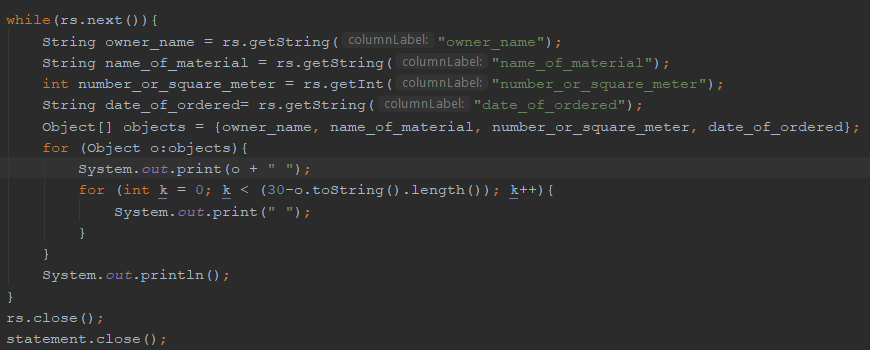
The method is same as method “print\_building\_materials”

Here it’s just opened database “ordered\_material.db”

And the we create statement in order to execute our simple SQLite queries. Then there is the variable “query” that contains query to our statement. And variable rs(ResultSet) gets results from statement.executeQuery(query).

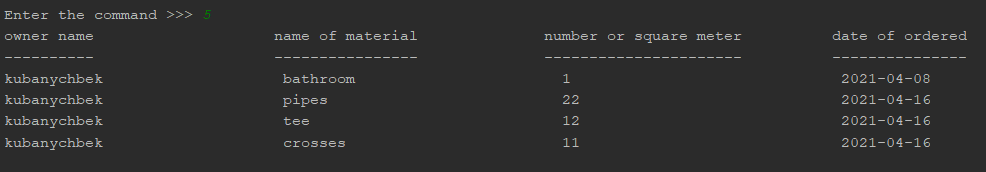
Name of columns:



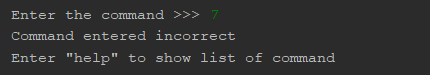


And here the program will print values through loop(while). There is also another loop(for) of our primary loop, with helping of this loop(secondary), our values will be printed in straight.

Like here:

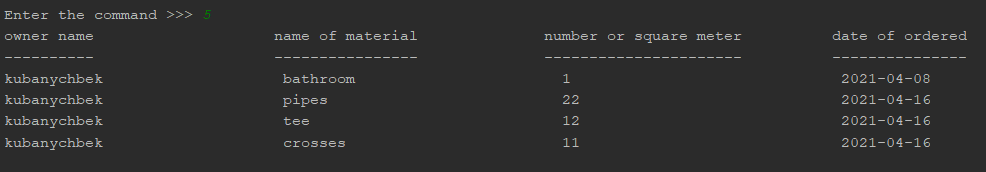


If user enters incorrect number of command, there will be appeared message:



To log out from the account of owner, user enters “0” and user logs out





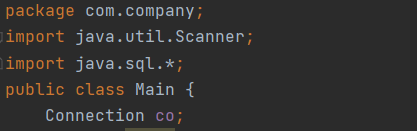
# Deliveryman’s menu

The deliveryman, in fact, is the main executor of requests from the client, so we tried to integrate useful and visual functions into the menu that would help in the work and provide access to certain functions.

The delivery service has a number of functions. For example, the deliveryman can view the list of materials for delivery and the list of delivered items, as well as deliver the goods recorded in the "ordered materials".

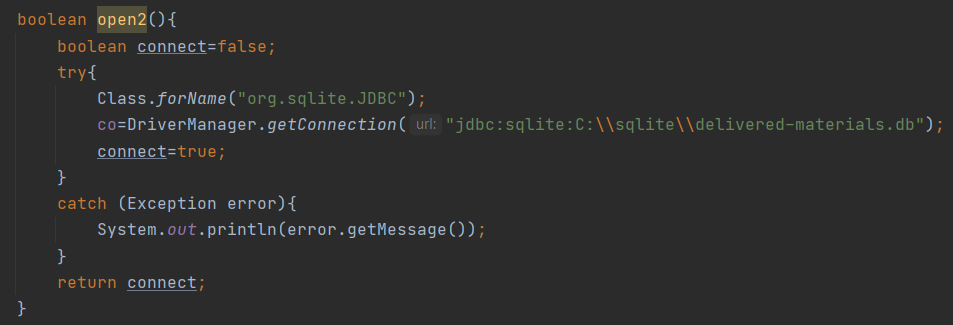
The code structure consists of seven code arrays (six of them are the main functions, and the last one will allow you to exit the menu).

The file starts with importing useful library: sqlite3. Then we attach the file .db and create an object that will allow us to interact with the database:



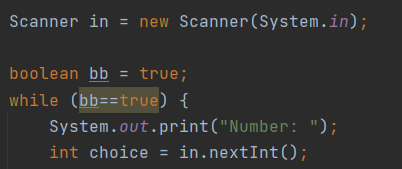


With this method, we connect and open our database named “Ordered materials”.

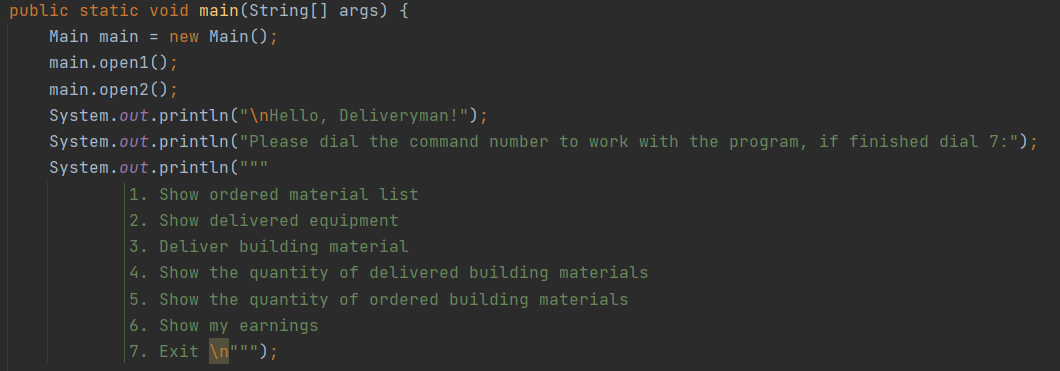


With this method, we connect and open our database named “Delivered materials”. We decided to create 2 methods, so we do not have to enter the same code every time.

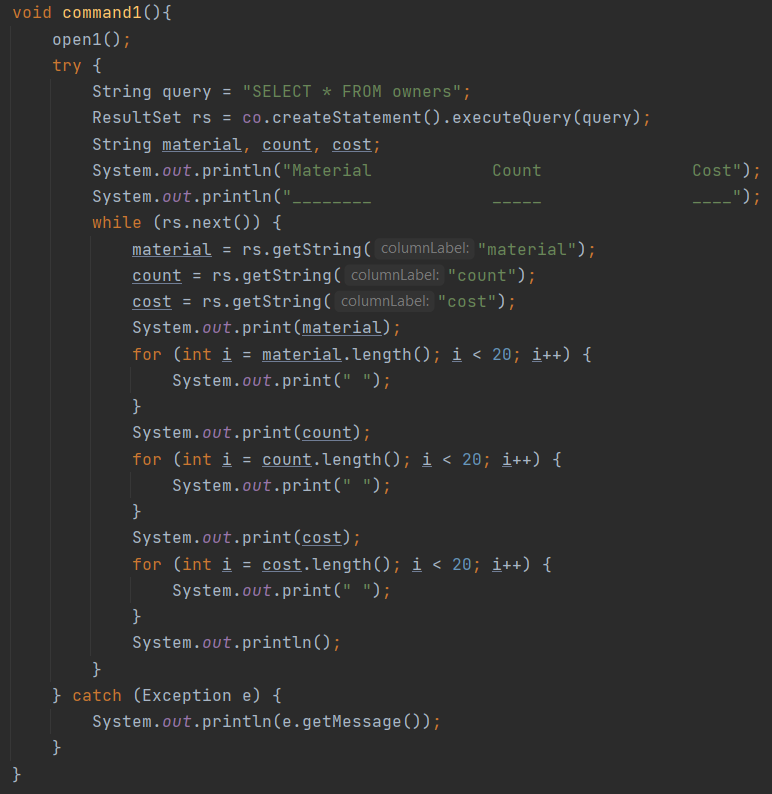
We include Scanner that take in user’s data. It allows choosing one of seven commands (functions).



At the beginning of our program, the deliveryman's greeting and a menu with 7 items are displayed on the screen:



1. There is the first command that show all ordered material list:

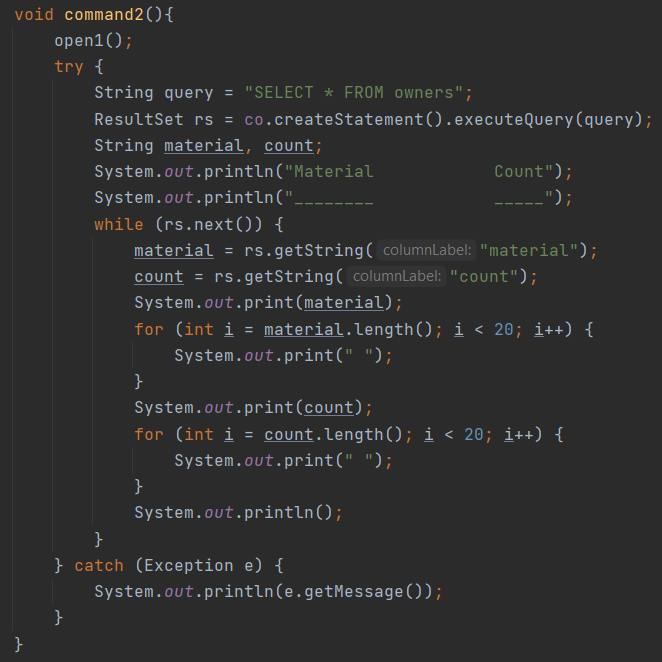


Firstly, we connect to our owner’s database. Then we create String variable to create table with “material”, “cost”, “count”. All this actions we execute with WHILE Loop (if it has next lines) and it consist of FOR Loops. All code written in “try/catch” (if we would have error).

“Ordered\_material” is the database with information about all materials that clients ordered. It has such table with:

* + - Material – name of all materials;
    - Count – material’s quantity;
    - Cost – the price of materials.

1. There is the second command that show delivered equipment:

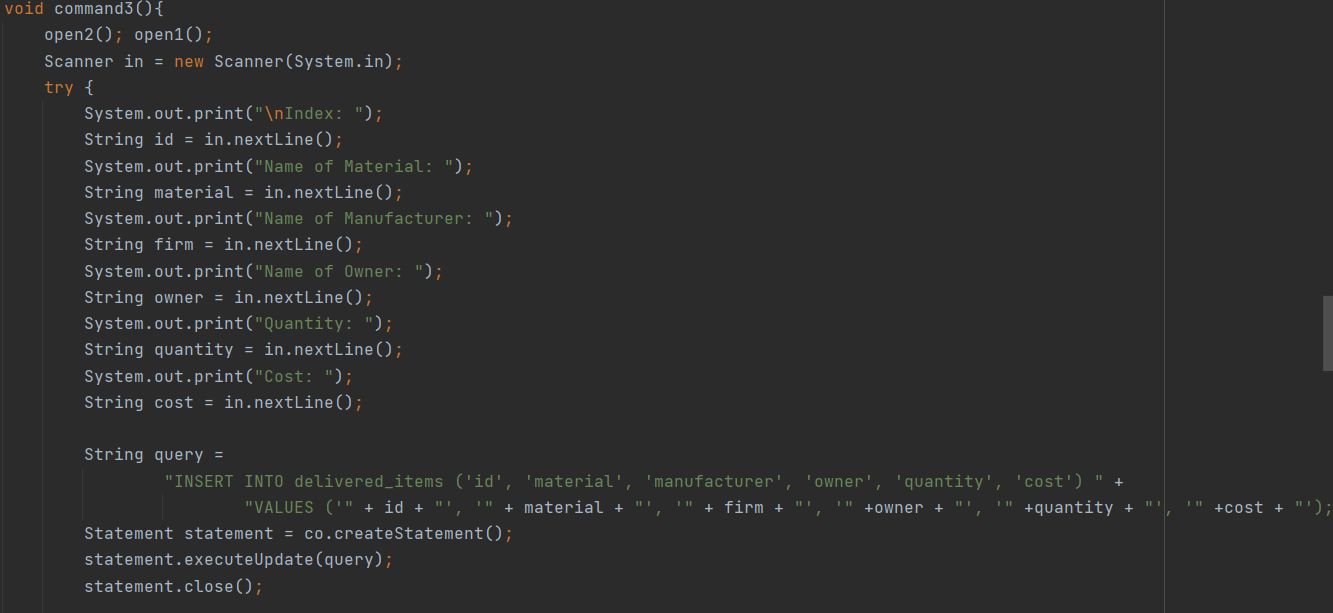


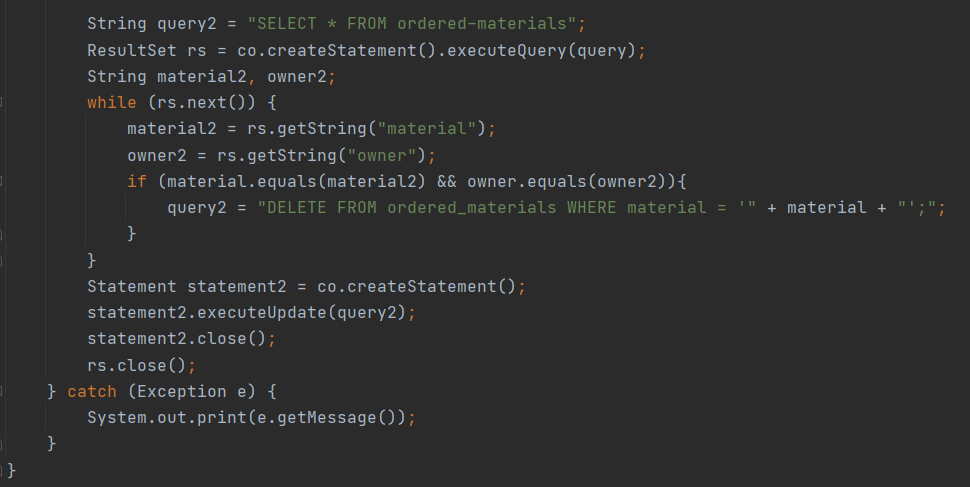
Our second command allows us to see equipment that delivered to clients.

“delivered\_equipment” is the database with all confirmed orders for delivered equipment. It has table with:

* material – material’s name;
* count – material’s quantity;

1. This is our third command that allows enter data about material to deliver:



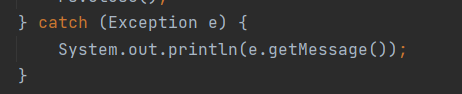


Firstly, we use methods to open two databases. Then program ask for information such as ID, name of materials, name of manufacturer, name of owner, quantity and cost. Then program will save all given information in table, also in database SQLite (row).

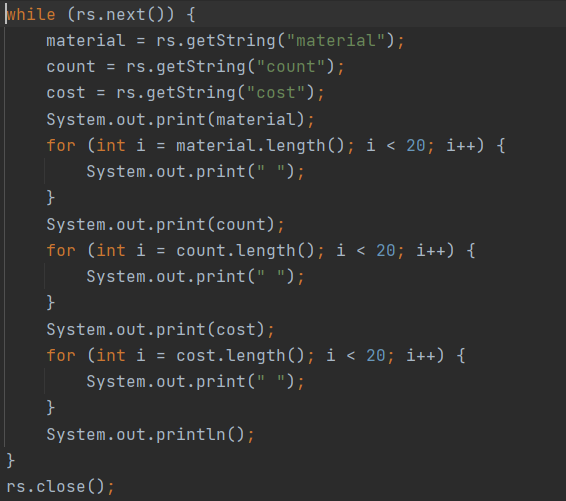
1. This is our fourth command that allows to see the quantity of delivered building materials:

Firstly, we are connecting to the database with open2() method. Then we use try/catch.



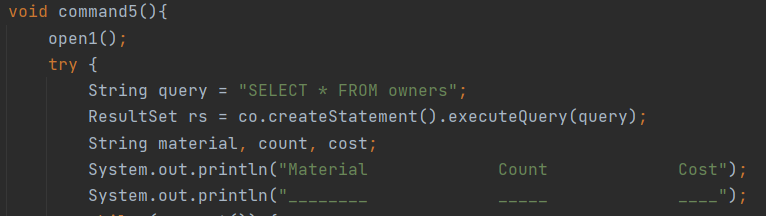


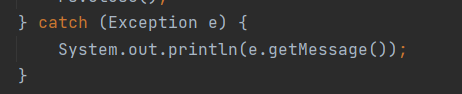
With WHILE loop and FOR loop we print columns with materials, count and cost. And then we obtain the quantity of delivered materials



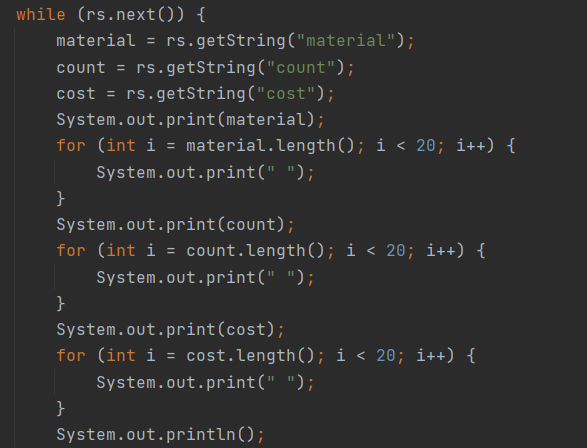
1. This is our fifth command that allows to see the quantity of ordered building materials:

Firstly, we are connecting to the database with open1() method. Then we use try/catch.



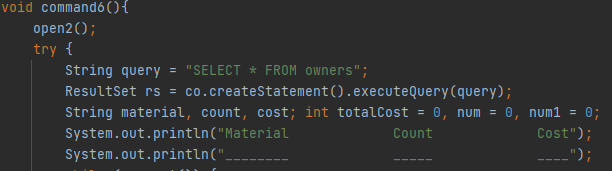


With WHILE loop and FOR loop we print columns with materials, count and cost. And then we obtain the quantity of ordered materials

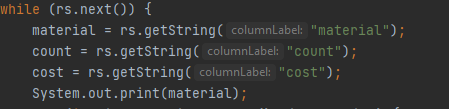


1. In the 6 command, we are connecting to the database with open2() method:

After we have connected to our delivered-items database we are selecting the whole table from database with the ResultSet Function and open the variables for our future code

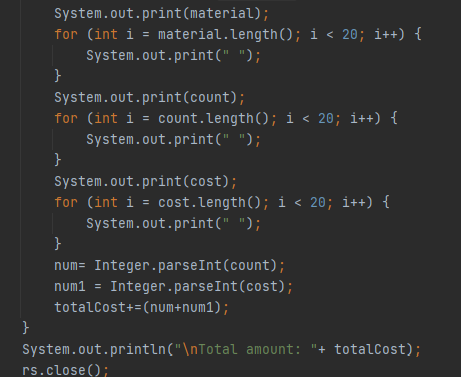


And when we open we have rows in database we use while loop and give this values for our variables.

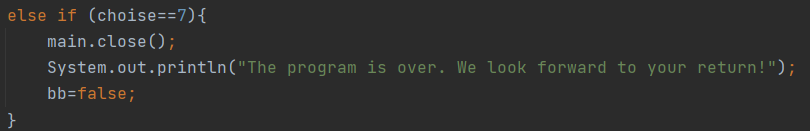


Then we print our values and with for loop we are printing space bars then we parse the String values to the Integer to keep our values in num to calculate them in future

In the end we print the Total amount of materials.

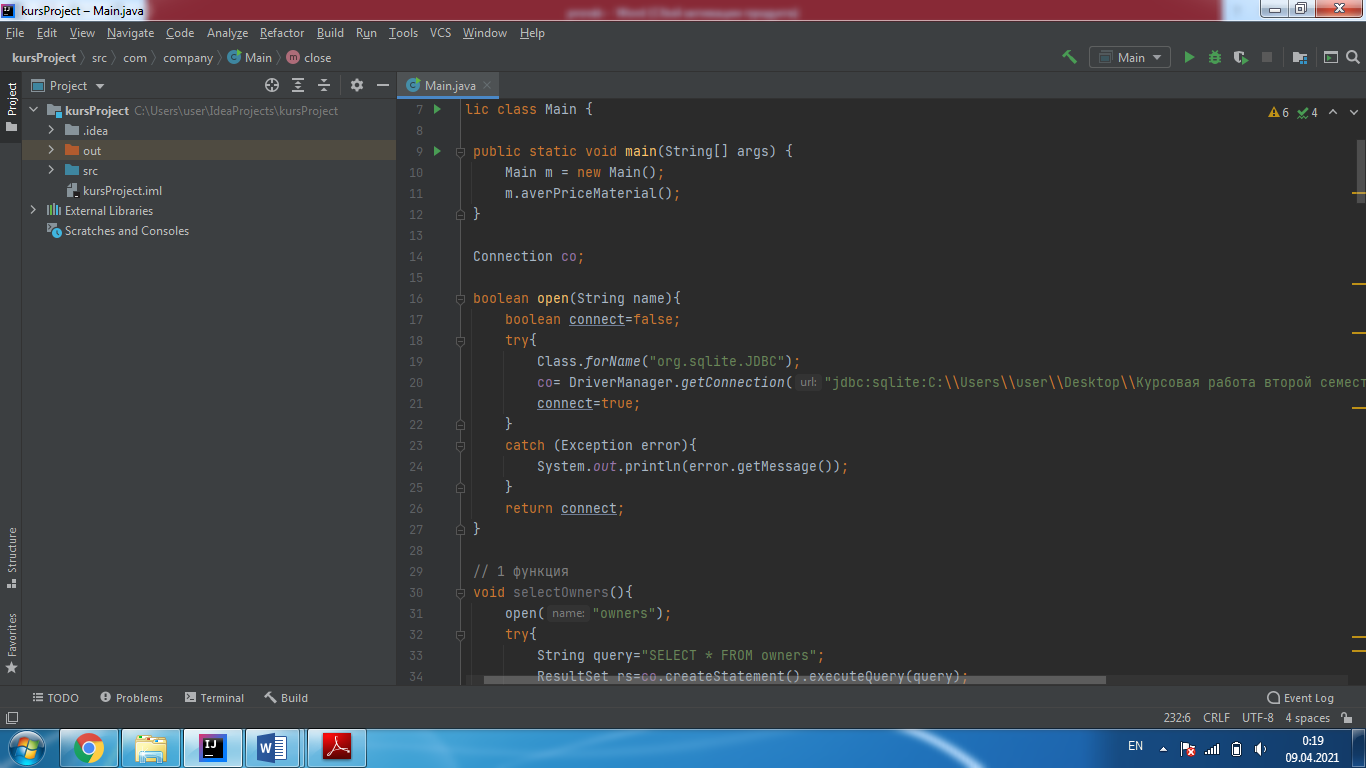


1. And when we want to exit from delivery’s program, we should enter the number 7. After that, our “infinite” loop closes.



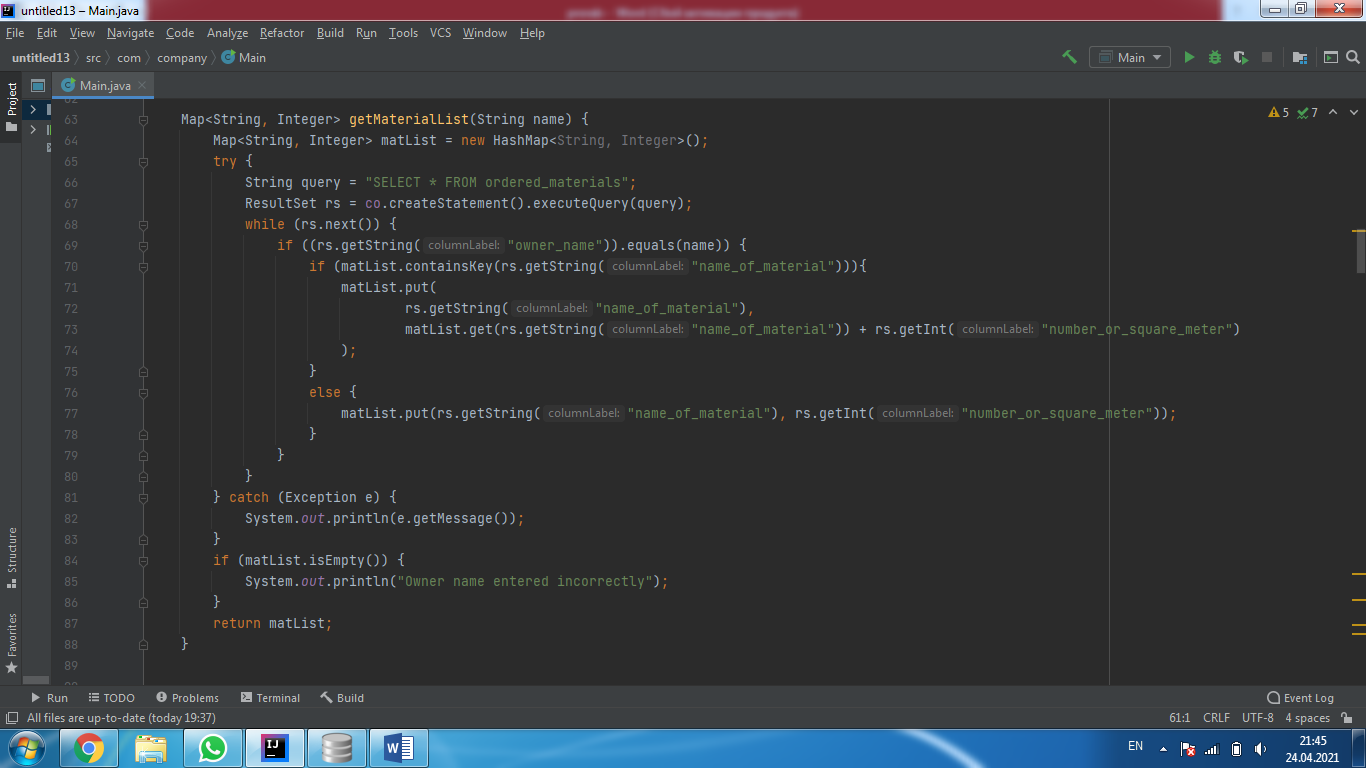
# Account of foreman

Let's start first with methods that help you implement commands and simplify your code.

The first method is called “open” and is used to open the database.

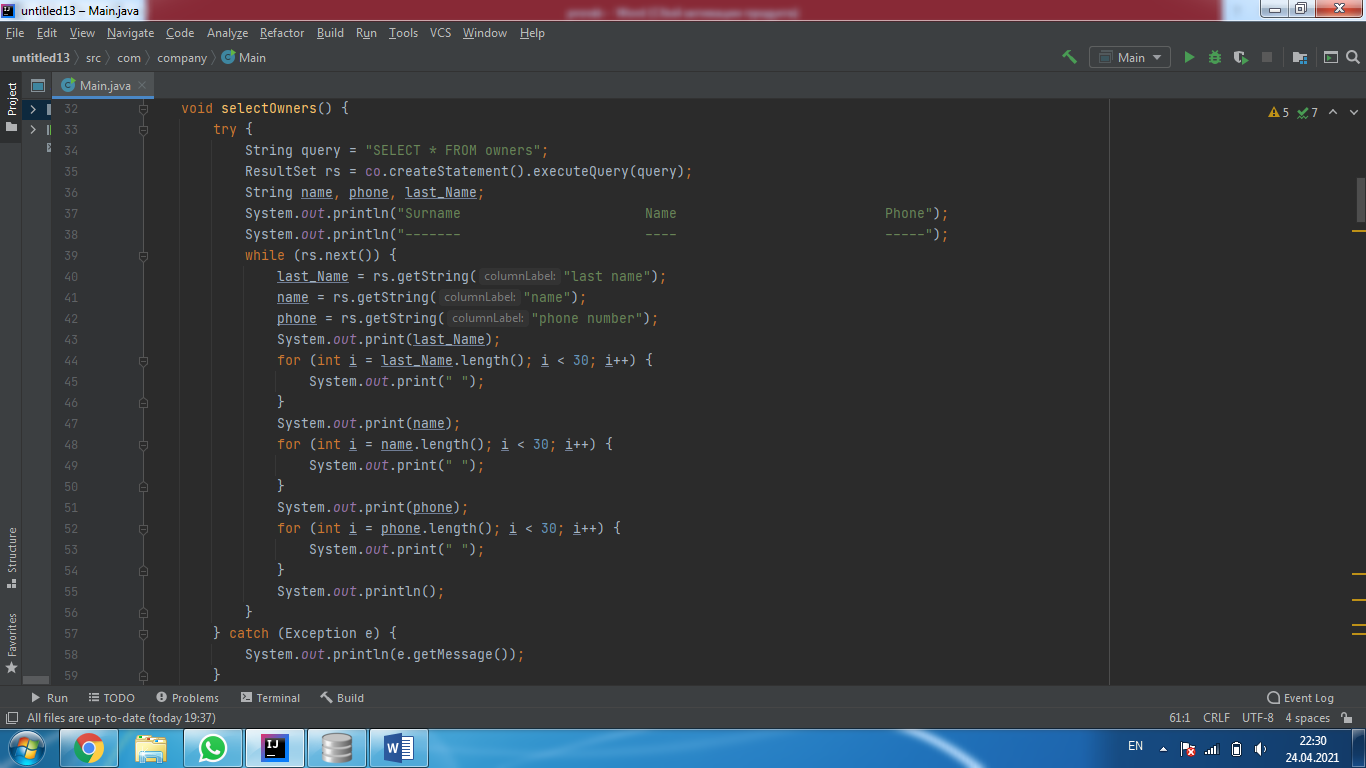
For input, it takes the name of the database to open and outputs the value true or false. The standard script for opening the database was used for writing.

The second method is called “getMaterialList”. The method is also very important, it returns a dictionary with the user's materials and their number, whose name you entered, also the method is used in 2, 3, 4 and 5 functions.



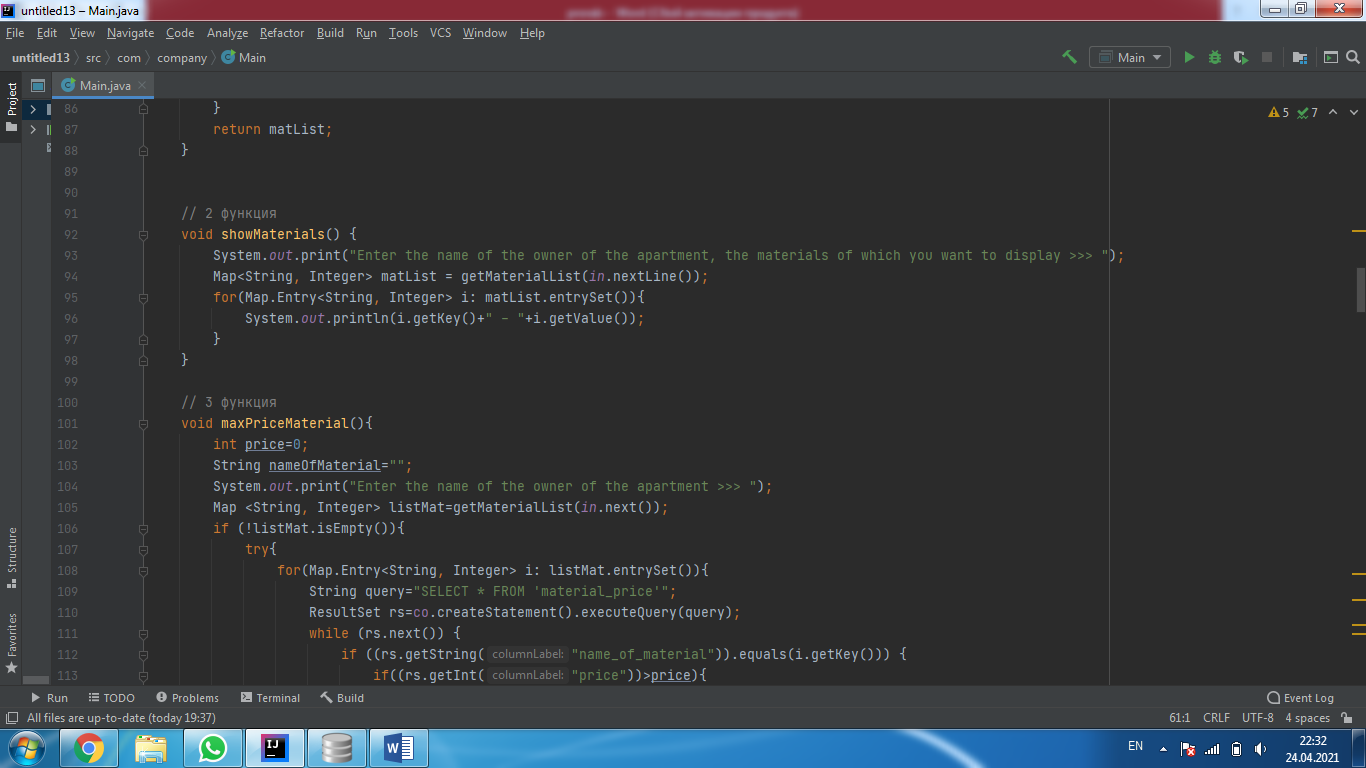
First, a “matList” dictionary is created with the key “String” and the value “Integer”. Then the program goes through the table “ordered\_materials” and checks the column “owner\_name” with the input parameter “name”. If they match, then the name of the material is added to the dictionary as the key and the amount of material with the given name as the key value. If the name of the materials is repeated, then a new value is added to the previous key value, that is, the amount of material increases. At the end, the program outputs a dictionary. Also, if this dictionary is empty, the program writes that the user name is entered incorrectly. Since if the dictionary is empty, it means that the program could not find such a user in the “ordered\_materials” table.

The first command is used to show a list of owners with their data. The name of the method of the “selectOwners” command.



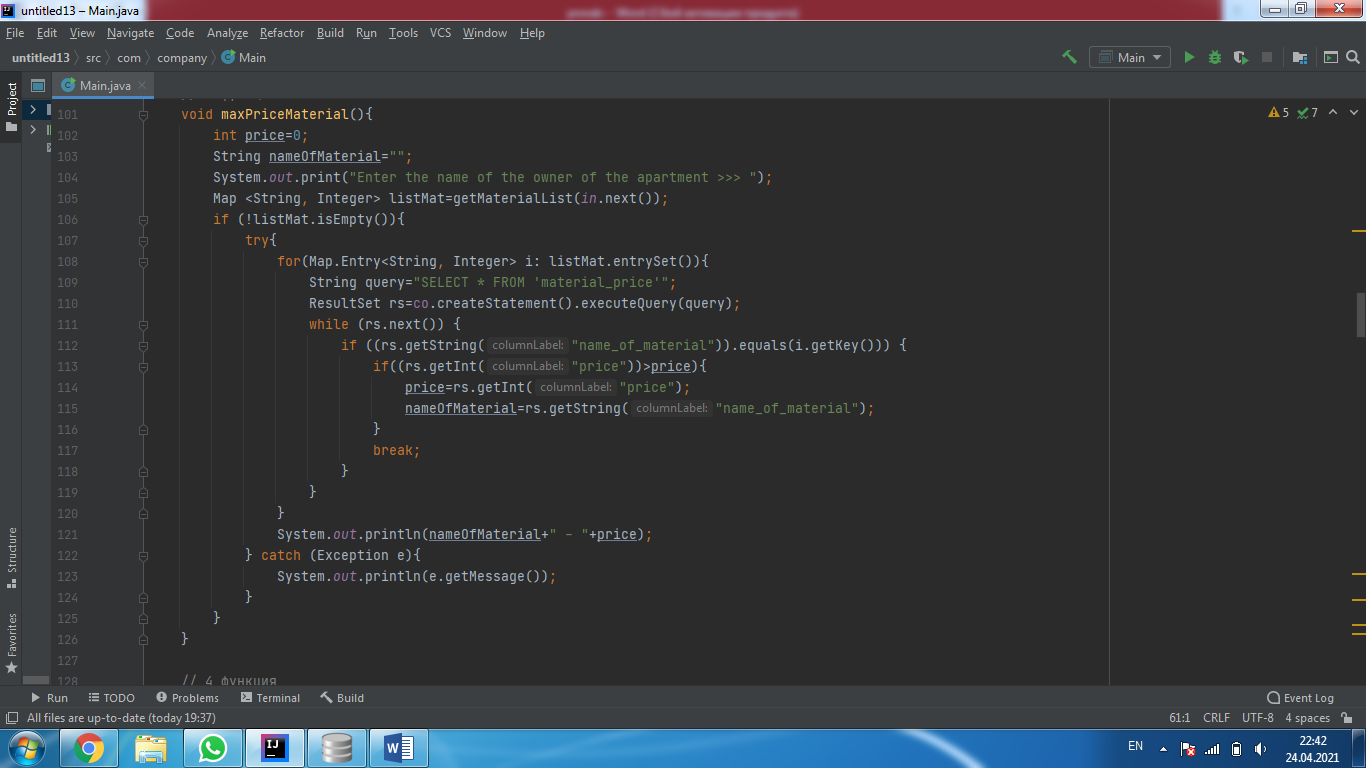
The first method uses a “while” loop and outputs the owners first name, last name, and phone number in columns from the “owners” table. Each column is 30 spaces wide, so the program calculates the length of each value taken from the database and adds the missing spaces using a “for” loop. This is done so that all the data comes out nicely in columns and is easier to read.

The second command is used to show the list of ordered materials of a certain owner. The method of the command is called “showMaterials”.



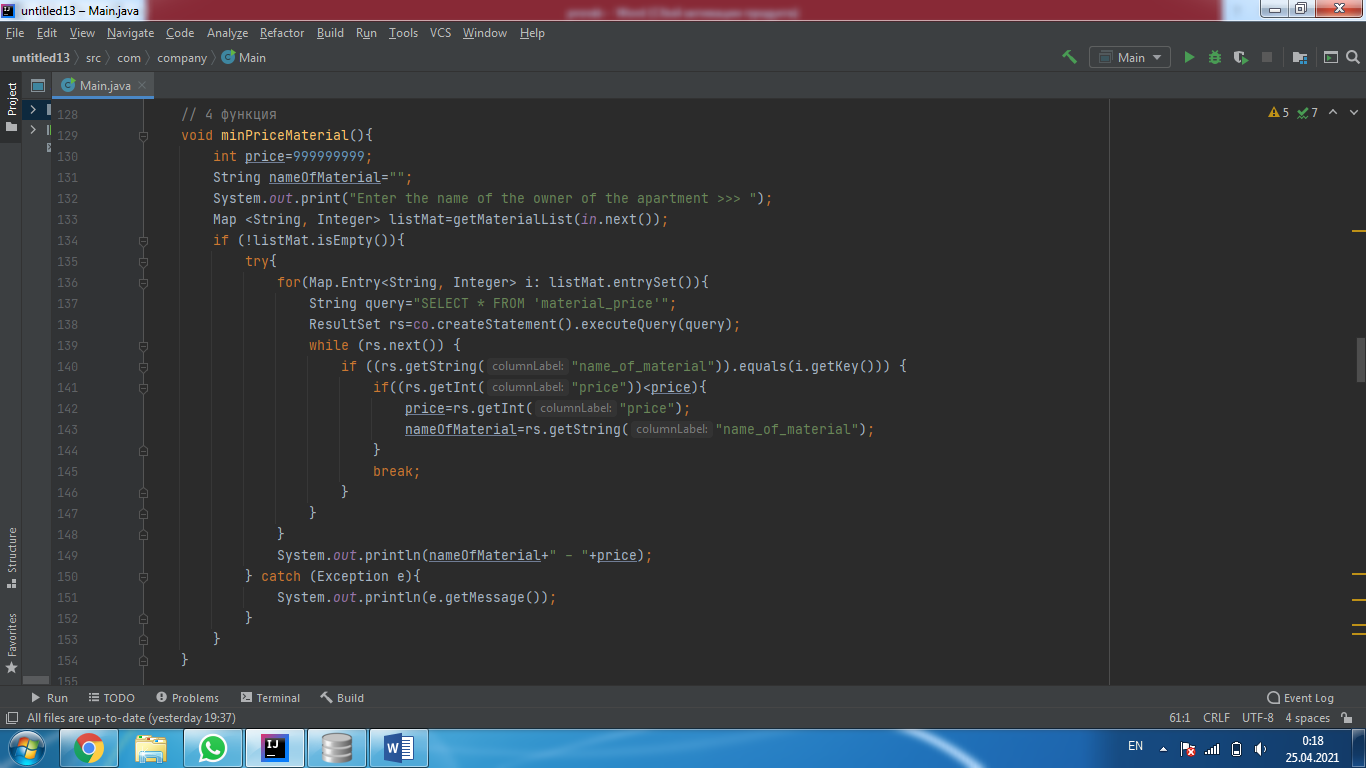
The method is very simple, with a size of five lines. First, this method asks you to enter the owner name to show a list of its materials. Then, using the “getMaterialList” method, it gets a dictionary with materials and their number and simply displays the entire dictionary on the screen using a loop. The method turned out to be very simple only thanks to the additional method where the main actions take place, this method only needs to display what is received on the screen.

The third command is called “maxPriceMaterial” and is used to find the most expensive material of a certain user.



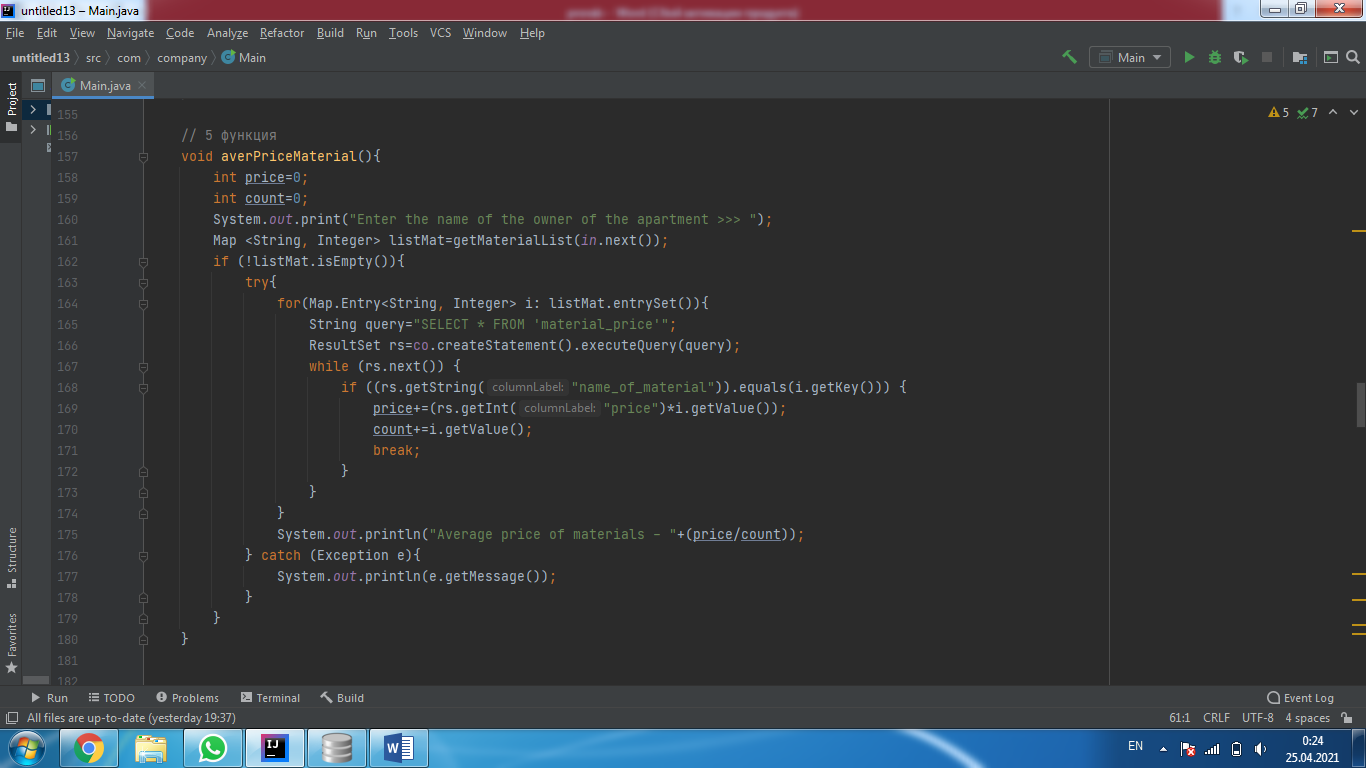
First, you create a “price” variable with the minimum value and a “nameOfMaterial” variable to store the name of the most expensive material. Then the method asks you to enter the name of the owner to get a list of its materials. Then, using the “getMaterialList” method, it gets a dictionary with the materials and their quantity. There is also a check, if the resulting dictionary is empty, then the code does not work further. It goes through the dictionary and uses a loop to read the table “material\_price” if the name of the material from the dictionary is equal to the name of the material from the read line, the price of this material is taken from this line and compared with the variable “price”. If the price of the material is greater than the variable, the variable becomes equal to the price of the material from this line, and the variable “nameOfMaterial” becomes equal to the name of the material from this line, and so on until the materials in the dictionary run out. Since the “price” variable will always be updated and grow, until it becomes the maximum and is no longer updated, and the “nameOfMaterial” variable will retain the name of the material with this price. This way we will get the name of the material with the highest price and its price. And we will output them at the end.

The fourth command is called “minPriceMaterial” and finds the material with the lowest price.



This method has the same scheme as in the “maxPriceMaterial” method, only now the “price” variable has the maximum value. And equate it only to values that are less than it. So we get the name and price of the cheapest material of a particular owner, and display them on the screen.

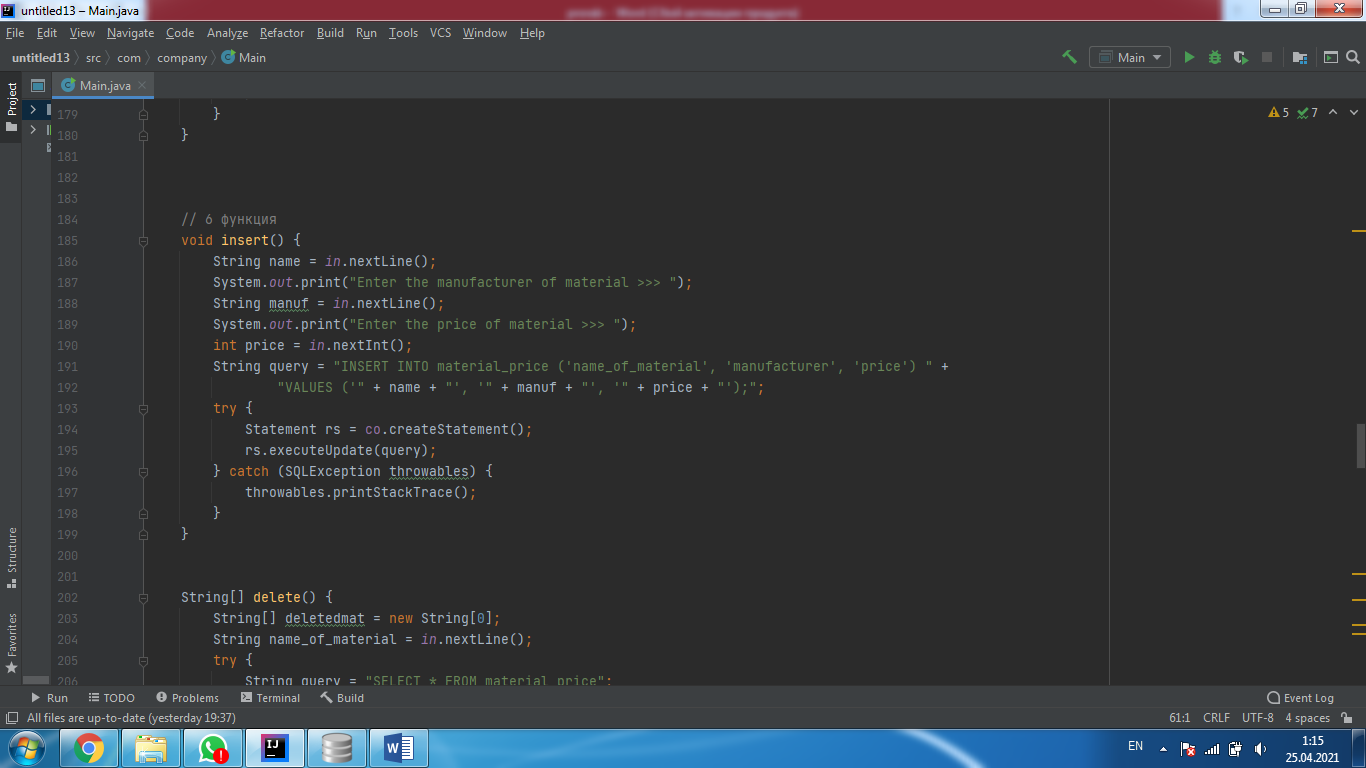
The fifth command is called “averPriceMaterial” and shows us the average price of all materials. Unlike the previous two commands, this method takes into account the amount of materials.



First, two variables “price” and “count” are created. Then the program asks you to enter the name of the owner and immediately creates a dictionary with the names of the materials and their number. The method goes through the entire dictionary, and for each element of the dictionary, it goes through the “material\_price” table using a loop. If the program finds material from the dictionary in the table, the method adds the price of the material multiplied by the number of these materials and the dictionary to the “price” variable. Also, at this time, only the number of these materials from the dictionary is added to the “count” variable. And so after we have run through all the dictionary values, we get two variables “price” with the sum of all prices and “count” with the sum of the quantities of materials. To display the average price for materials, we divide the amount by the quantity, that is, we divide “price” by “count” and display the result.

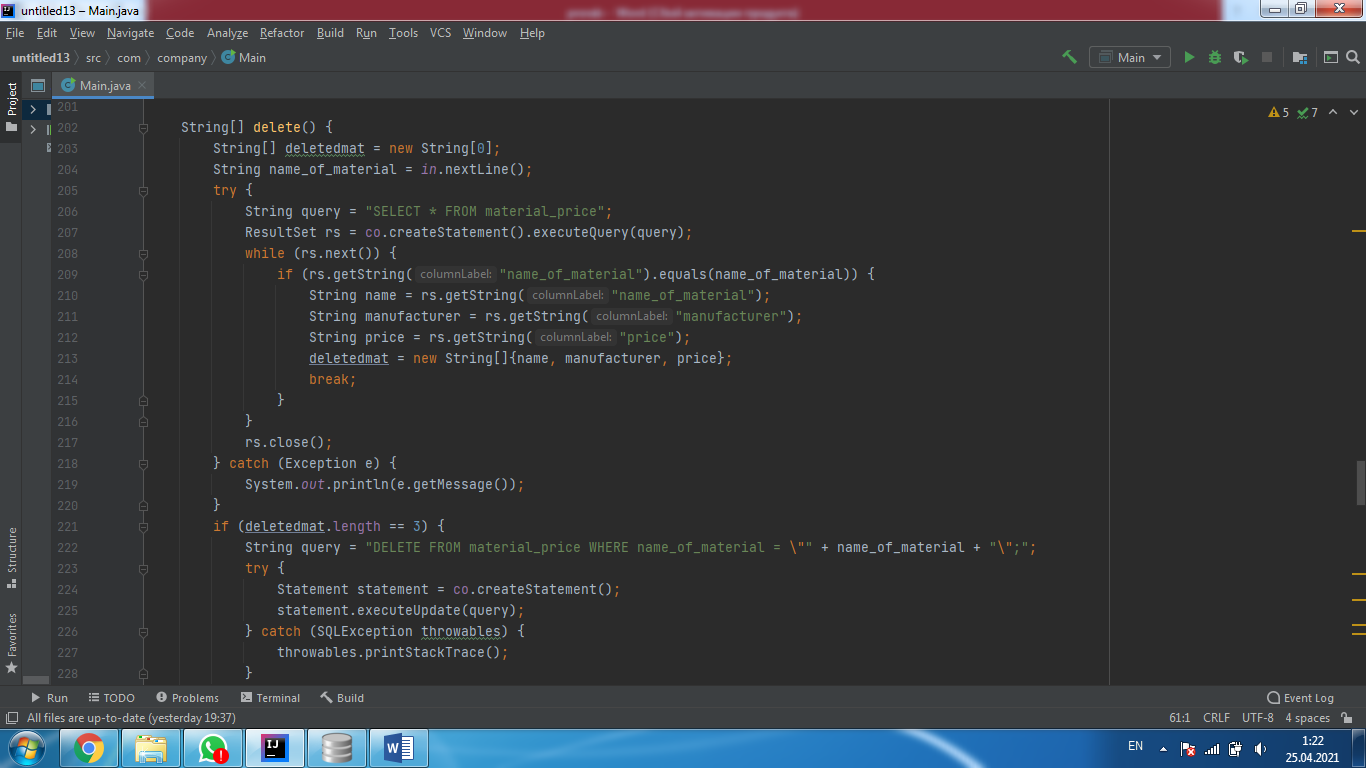
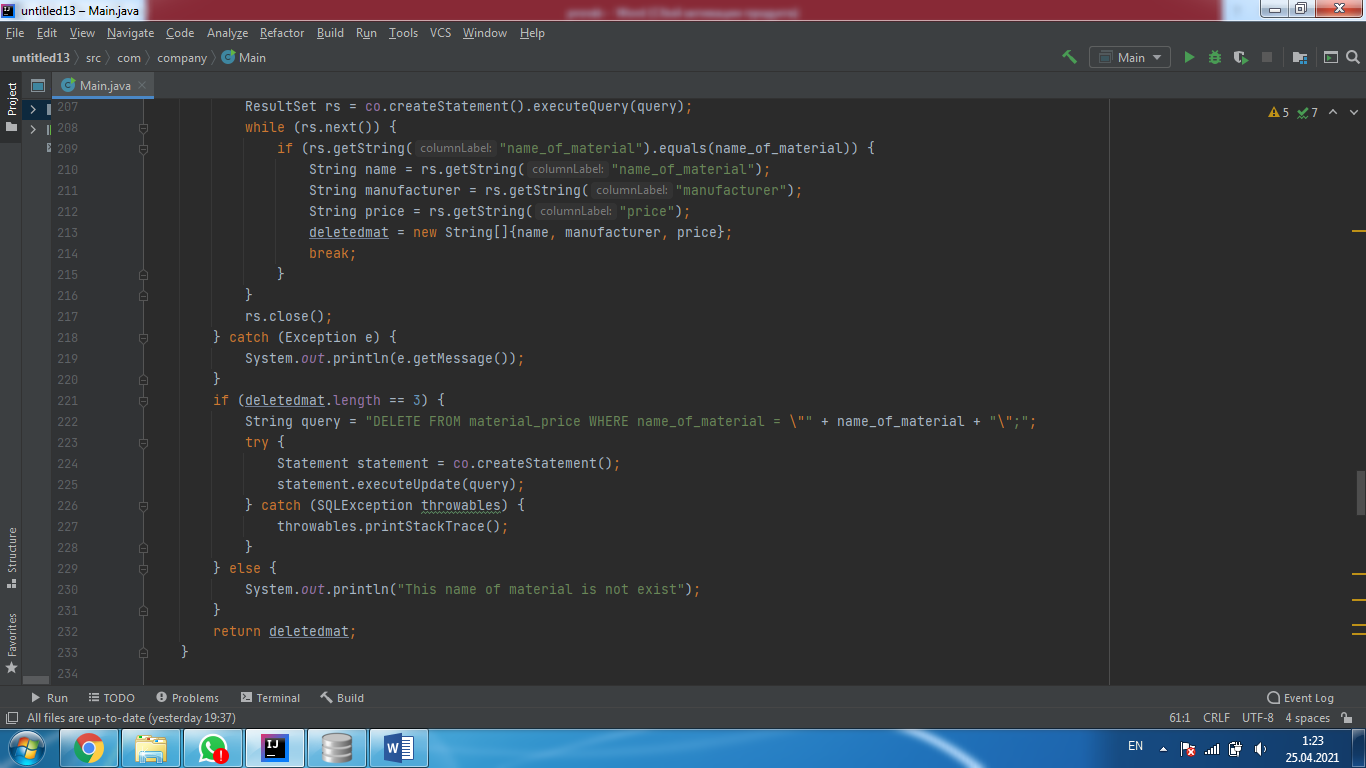
In the sixth command, there are three methods “insert”, “delete” and “update” at once, they are used to add, delete and change values in the “material\_price” table.

First, consider the “insert” method for adding materials to a table.



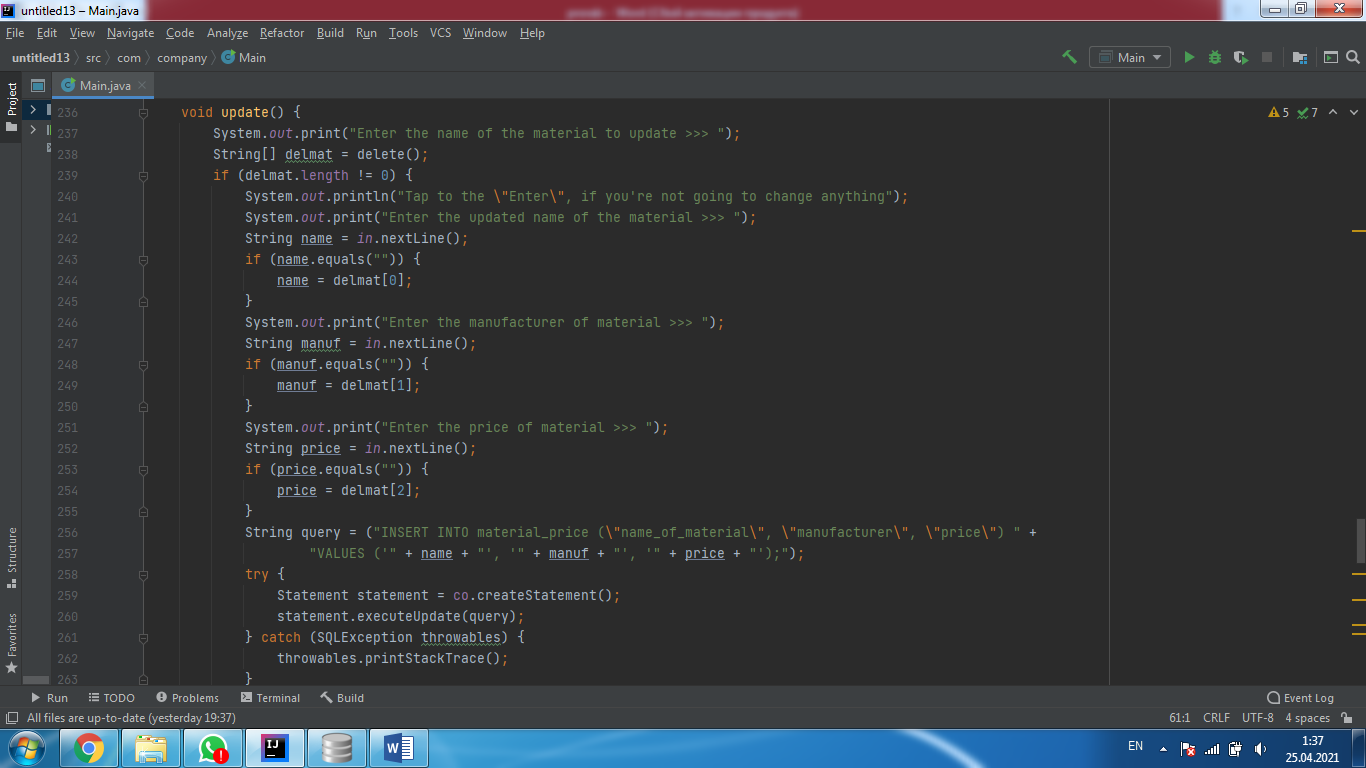
At the beginning, the method requests the name of the material, the name of the company, and the price for the material and stores them in variables. Using these variables, the method uses the standard command from sqlite to add a row to enter the values of the variables in the “material\_price” table.

The next method is “delete” to delete a certain material.

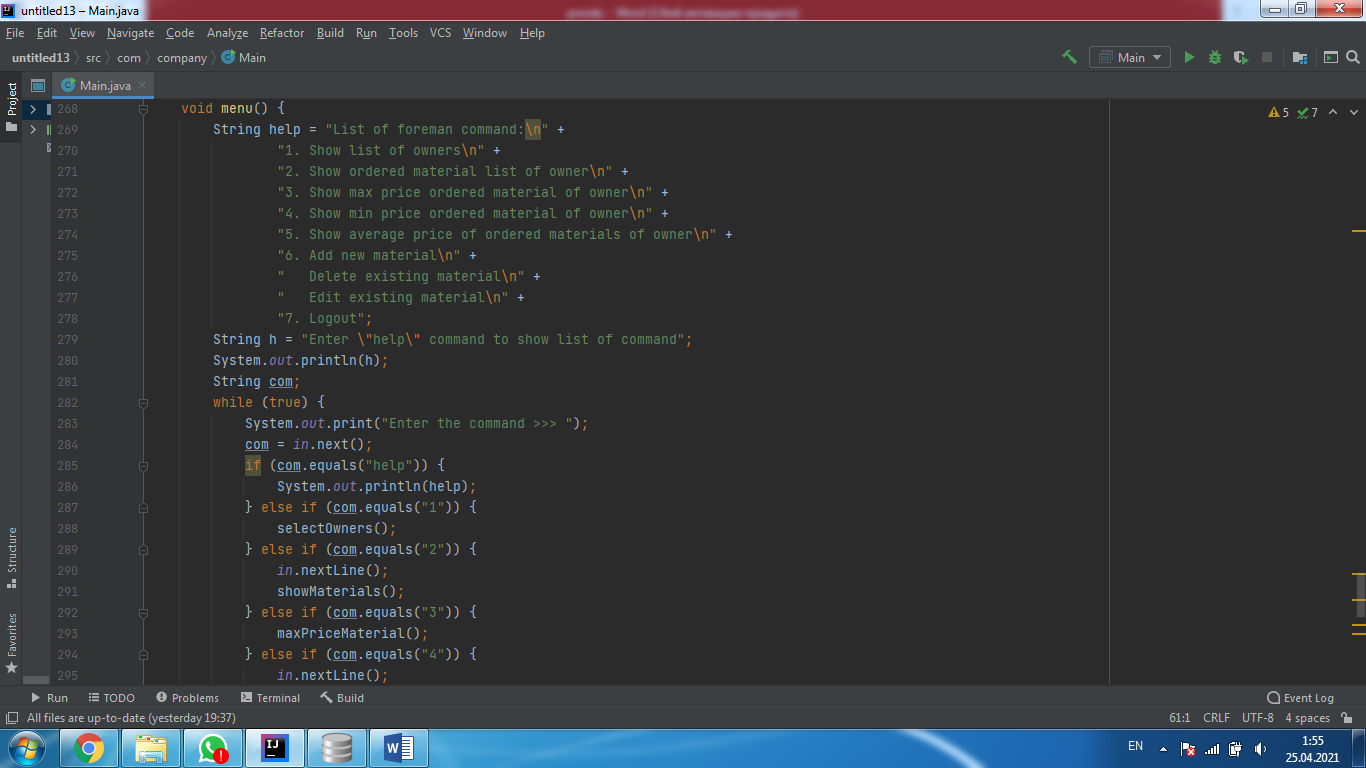
The method starts by creating a deletedmat list variable to save deleted content. Then the program asks you to enter the name of the material to delete. After entering the name of the material, the program starts searching for this material in the “material\_price” table. If it finds this material, it writes the name of the material, the company name, and the price to the “deleted\_mat” list variable. Then the program will check whether something was written to this variable, if so, the program goes on, if not, the program prints that such material is not in the table. This is done so that the user understands that he deleted the material, because if you try to delete the material that is not in the table, the program will continue to work and the user will not understand whether he deleted it or not. After saving the deleted materials, the program uses the sqlite command to delete this material from the “material\_price " table. And outputs the list of deleted materials “deletedmat” outside the boundaries of the “delete” method.

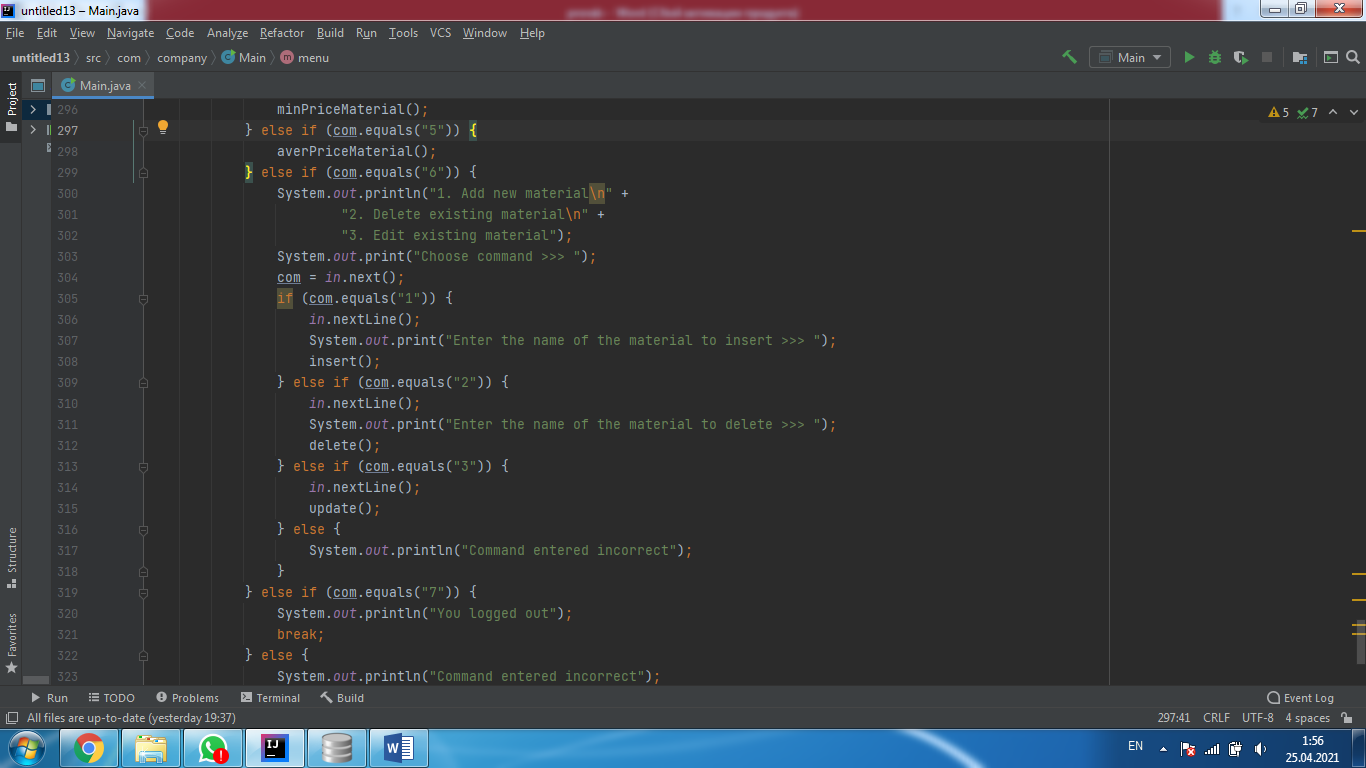
The last method is “update” it changes the data in the “material\_price” table.

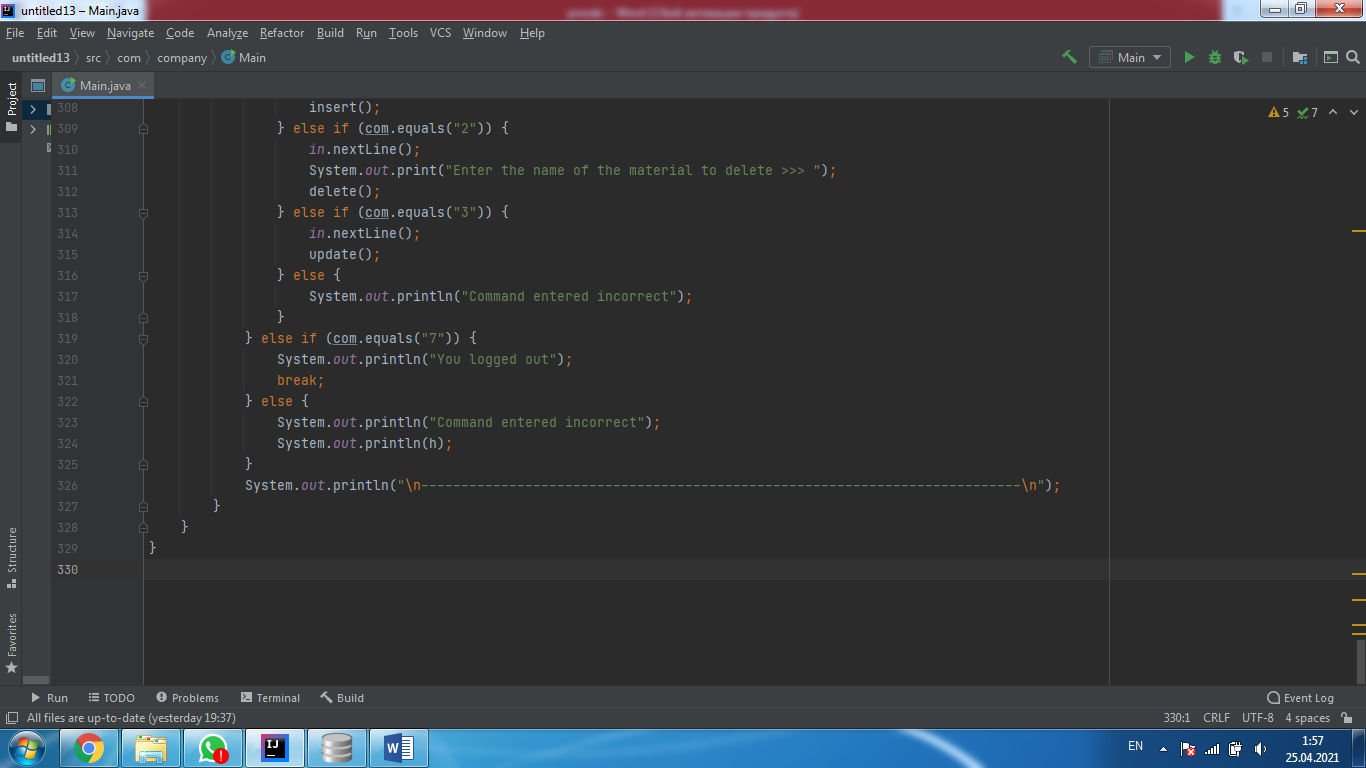


First, the method asks you to enter the name of the material that you want to change. After that, it immediately creates a “delmat” list and removes the item with that name from the table. Then it checks for the presence of items in the resulting list. If the list is not empty, the program continues and asks you to enter a new name for the material, the company name, and the new price. If you just press “Enter” instead of typing, the program will record the past value from the table and the variables to enter. This is done using the variable “delmat” obtained using the “delete” method, all this simplifies the program, because if you, for example, want to change only the price of the material, you can skip the values that you do not want to change and enter only a new price for the material. Then you don't have to enter the same material and company name again. After you have entered the new changed data, the method uses the variables in which you saved this data to write the changed material to the “material\_price” table.

And so there was the last method “menu” which combines all the previous methods into one.







Using the “while(true)” loop, the method starts an infinite loop for entering commands (1-7). After entering a command, the method uses “if, else” to determine which command the entered value belongs to and calls the corresponding method. If the command is entered incorrectly, the method gives a hint that you can enter "help" to view the list of commands. As you know, there is still a seventh command that serves to log out of the system, if you enter this seventh command, it will simply stop the loop, which will mean that the program has ended.

This is the end of the description of the foreman's account.

# Conclusion

The result of the work done on the introduction of automation of the information system in the selection of materials for the repair of apartments were the following results:

1) The work has become more interesting;

2) The quality of the work has improved and it has become more competitive;

3) Improving productivity and business sustainability;

4) Order processing has become much faster;

5) With the help of system automation, it is now possible to save time;

6) Coordinated work has been established between the clients, the foreman and the deliverers;

During this course project, our team got better at writing code and even learned new features. Each participant made a contribution. We really worked as a team.

Our work was hard, but at the same time we were interested in doing it, and it was useful for the programmer to write codes. We learned how to work better with SQL, connect multiple codes, and maintain code documentation. We have shown you how to start a company that is engaged in the selection of materials for the repair of an apartment

# Literatures:

SQlite java: <https://www.youtube.com/watch?v=O5UpLpekpdw&list=RDCMUCn5VAW5rLUZx3lXSgwTlBMw&index=3>

<https://www.youtube.com/watch?v=XnWYNIMTRHY&list=RDCMUCn5VAW5rLUZx3lXSgwTlBMw&index=1>

<https://www.youtube.com/watch?v=Eta9knRFJLE&list=RDCMUCn5VAW5rLUZx3lXSgwTlBMw&index=2>

Askar Almazbekovich - Java course