This document is a set of instructions on how to code an ATM using Java. It teaches the basics of coding.

Creating an ATM in Java

Learn a Simple Code Program

Brandy Wingfield

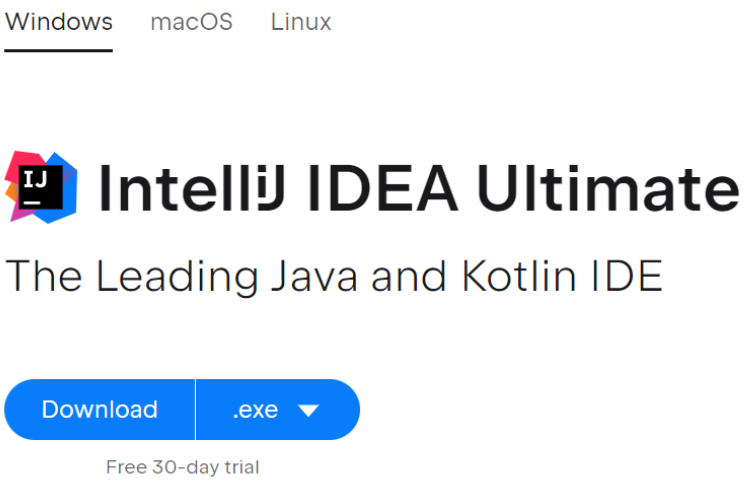
Contents

ATM Program Introduction

If you ever wanted to learn how to code, then these set of instructions will teach you. You will learn how to code using in an exciting way because you will be coding an ATM. You will be working with IntelliJ IDEA to code your ATM in Java. As you are coding, you will have a better understanding of

Getting Started

Before starting the ATM program, you must have a computer software to run your Java code. In this set of instructions, you will be working with IntelliJ IDEA. To install go to https://www.jetbrains.com/idea/download/?section=windows.



Select a Version

Select the version that works on your device, which will be either Windows, macOS, or Linux. After, scroll down to download the community version, and it would automatically download the file. IntelliJ IDEA should automatically open, but if not go into your downloads folder and open it manually. If you have macOS you would have to manually open the .dmg file, which is also found in the downloads folder.

Figure 1: Visual of the website IntelliJ IDEA with the types of versions

Windows Setup

To complete the setup for IntelliJ IDEA using windows, you will be shown a setup prompt, which will ask you if you would like to change the settings to your liking. Continue to choose next and it will eventually ask you to install. After installation, IntelliJ IDEA will open.

macOS Setup

To complete the setup for IntelliJ IDEA using macOS, you will need to access your downloads folder and drag the file to the applications folder. It will ask you a question regarding the terms. Accept the that you have read them and after you will be asked another question. IntelliJ IDEA will open after.

Lenox Setup

To complete the setup for IntelliJ IDEA using Lenox, you will open a new terminal and search “sudo snap install intellij-idea-community -- classic”. IntelliJ IDEA should now be successfully installed and opened.

Starting a Project

To start a new project, select “Create New Project”, and select "Java”. After select “Download JDK” in the drop-down menu. A prompt will show and select “Oracle OpenJDK” in the vendor section. After completing this step, click next until you are shown another prompt that tells you to enter your project name. Name your project and choose where you would like it to be stored on your computer. Your IDE should now be open. Next, right click the folder “src”, which is located on the left side of your program under the drop-

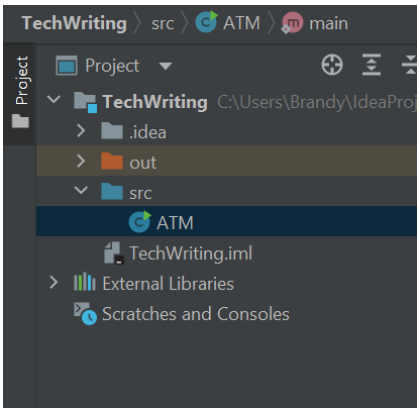


Figure 2: Starting Project

down menu of your project. Then, hover over “new” and then select “Java Class”. Name your program, “ATM”.

WARNING

Syntax in Java Programming

Syntax is important for your program to work. In Java, after each line of code the needs to be a semi-colon. So, treat the semi-colon in programming, like we do periods in sentences. Another important character that Java uses are curly brackets. We use curly brackets at the beginning and end of our loops, methods, classes, etc. If your syntax is wrong, your program will break.

Setting Up the Program

Scanner in Programming

Scanner is a tool in programming that we use to prompt the user to give information back to the computer. In this program we are going to use the scanner to ask the user a few questions related to depositing or withdrawing money from the ATM.

Step 1: Importing the Scanner

First, you are going to import the scanner package. To do this you are going to type on line 1 on your program, “import.java.util.Scanner;” and do not forget to end with a semi-colon.

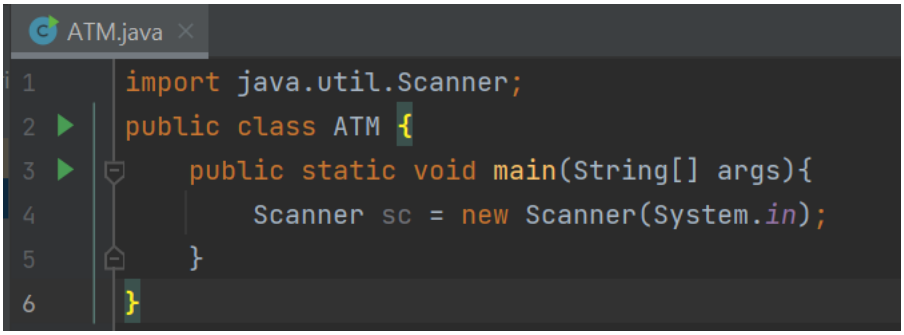
Step 2: Making your ATM Class

Classes are bodies of code that have a specific function. You are going to make an ATM class that has the properties of an ATM. To make the class, you are going to go to line 2 of your code and type, “public class ATM {”. Instead of a semi-colon you are going to use a left curly bracket, which is “{”. Go to line 3 and type, “public static void main(String[] args) {”, and use another left curly bracket.

Step 3: Bringing the Scanner to Life

Now that we imported the Scanner package into the program and made our class we now must initialize our scanner. Go to line 4 and type “Scanner sc = new Scanner(System.in);”. Do not forget the semi-colon. After, put a right curly bracket, which is “}”, on the next two lines.

Your code should look exactly like the image on the right.

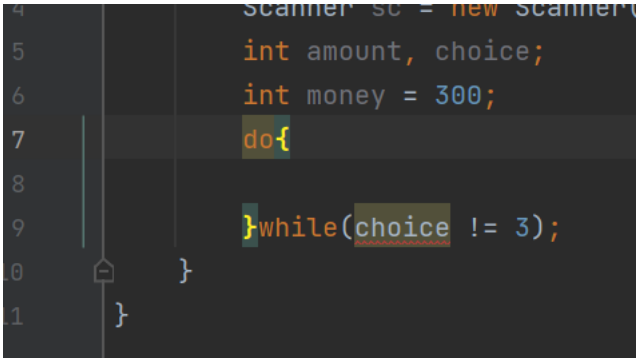


Step 4: Creating our Variables

Variables are used in programming to keep track of pieces of information. In the ATM program we are going to use three variables, which are money, amount, and choice. Money will keep track of the balance in the ATM, amount will be based on the user’s response on how much they would like to withdraw or deposit, and choice will be based on the user’s response if they want to withdraw, deposit, or leave the ATM. Go to line 5 and type “int amount, choice;” After going to line 6 and type, “int money = 300;” Do not forget the semi-colons.

Step 5: Loops

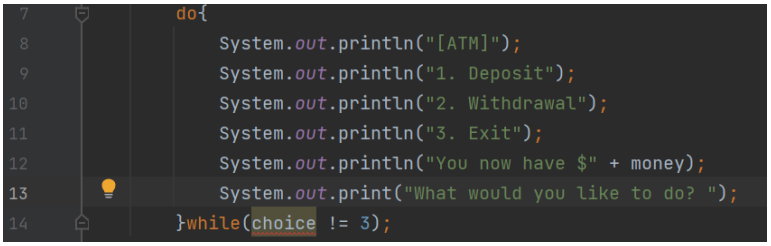
Loops are bodies of code that are repeated without writing more than once. The loop you are going to work with is a do-while loop, which will be repeated only if the user wants to make another withdraw or deposit. To start, go to line 7 and type “do{“. Use a left curly bracket. Skip a line 8 and on line 9 write “}while(choice != 3);”. Do not forget the semi-colon or right curly bracket.



Your variables and loop should look like the image on the right.

Step 6: Creating the ATM Menu

The code, “System.out.println();” is used to display information to the user’s screen. Write the code exactly as listed below and do not forget the semi-colons. Your code then should look like the image below.



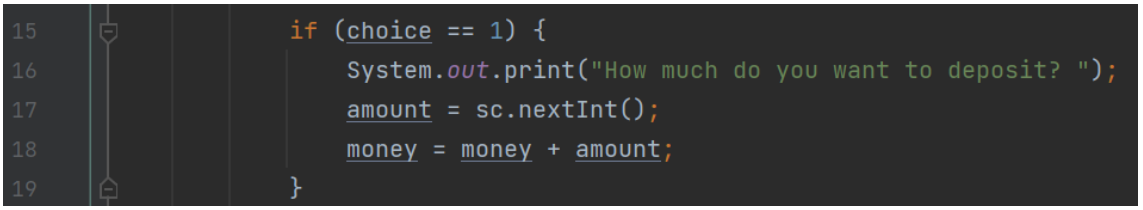
- Line 8: “System.*out*.println("[ATM]");”
- Line 9: “System.*out*.println("1. Deposit");”
- Line 10: “System.*out*.println("1. Deposit");”
- Line 11: "System.*out*.println("3. Exit");”
- Line 12: “System.*out*.println("You now have $" + money);”
- Line 13: “System.*out*.print("What would you like to do? ");”

Step 7: Using the Scanner

The previous print statement asked a question on what the user would like to do. Now we get to use the scanner. On line 14, write “choice = sc.nextInt();”. Do not forget the semi-colon.

Step 8: Condition Statements

Condition statements are used when we have multiple choices. The conditional statements in the ATM program will be based on the user’s choice.



Step 8.1: First Conditional Statement

Write the code exactly as listed below and do not forget the semicolons and curly brackets. It should look like the image above.

- Line 15: “if (choice == 1) {”
- Line 16: “System.*out*.print("How much do you want to deposit? ");”
- Line 17: “amount = sc.nextInt();”
- Line 18: “money = money + amount;”
- Line 19: “}”

Step 8.2: Second Conditional Statement

Write the code exactly as listed below and do not forget the semicolons and curly brackets.

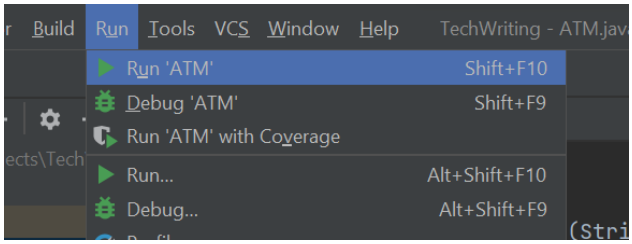
- Line 20: “else if(choice == 2) {”
- Line 21: “System.out.print("How much do you want to withdrawal? ");”
- Line 22: “amount = sc.nextInt();”
- Line 23: “money = money - amount;”
- Line 24: “}”

```
else if(choice == 2){
    System.out.print("How much do you want to withdrawal? ");
    amount = sc.nextInt();
    money = money - amount;
}
```

Your code should look exactly like the image on the left.

Let’s Test our ATM!

If you have coded everything up to this point, you now have a working ATM. To run your ATM, click the button, “Run”, located at the top of the screen, and then click “Run ‘ATM’”, which will be displayed as the first option in the drop-down menu.



You will then be shown a prompt down below, which will be your ATM program. You can now work with your ATM by answering the following questions that are displayed.

The image on the right is a test run.