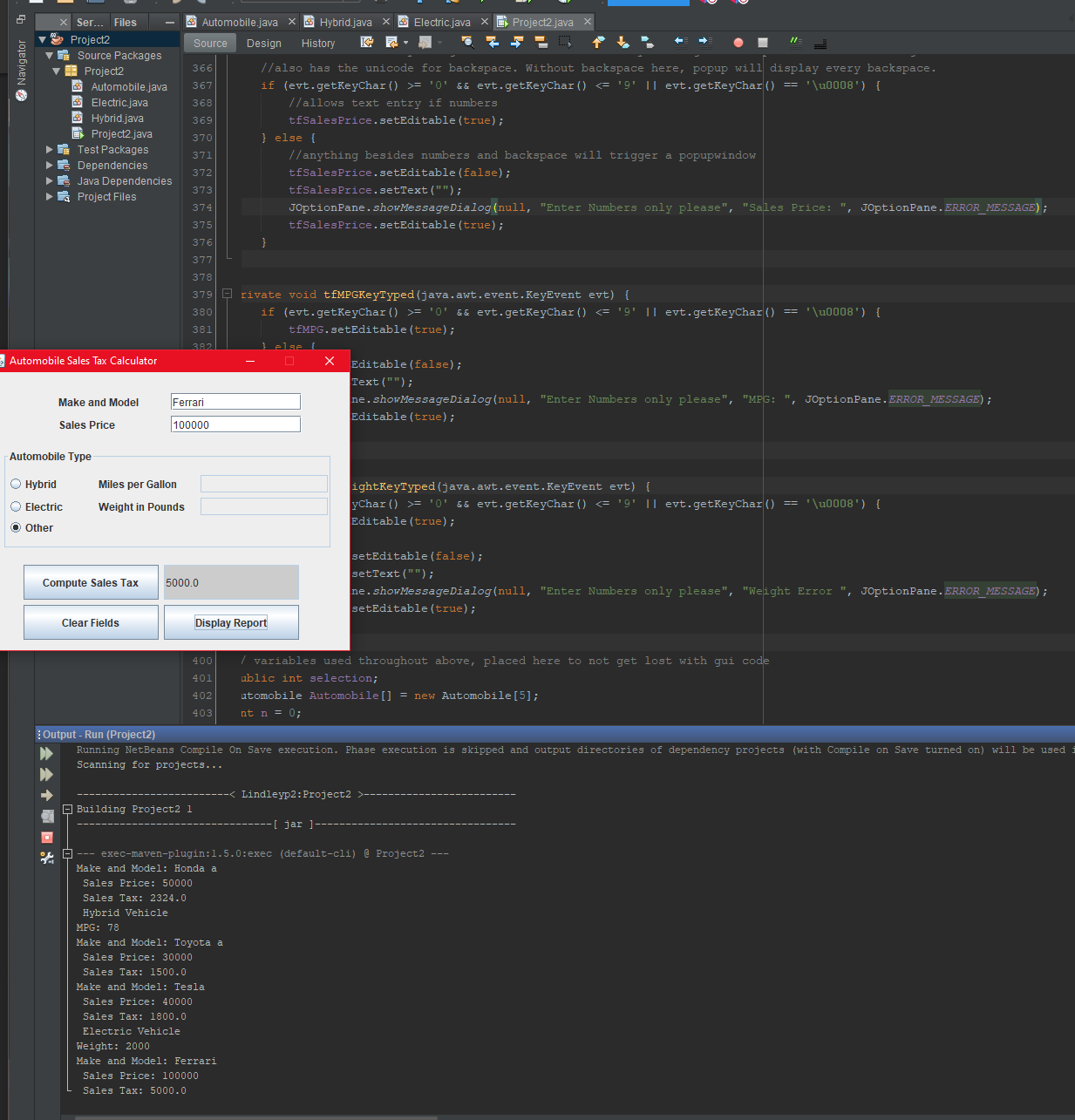
My solutions to the project:

* Automobile.java was straight forward. I have the variables declared as protected so child classes can access them. The sales tax method multiplies the sale price by 0.05 in order to get the sales tax of five percent. Also, the toString method here prints all the information.
* Hybrid.java is a child class of Automobile.java. However, hybrid accepts the argument of MPG and uses this to calculate sales tax, since it is dependent on MPG. The sales tax method for hybrids has an if statement for when MPG is less than 100 dollars and applies a 100-dollar discount to the five percent sales tax. The else statement handles MPG greater than 40 by subtracting 40 from the MPG and multiplying that by 2 dollars. This total is then added to the 100-dollar discount. The to string method here also adds the words “Electric vehicle” and displays MPG.
* Electric.java is like hybrid.java. It is a child class of Automobile, and the biggest difference from hybrids is that its discount is reliant on weight of the vehicle. It applies a 200-dollar discount to vehicles under 3000 pound and a 150-dollar discount to vehicles over 3000 pounds. This is achieved using an if statement for under 3000 and else statement for greater weights. The to string methods is similar to hybrid, but has information on the vehicles weight and that the vehicle is electric
* Project2.java is the meat and potatoes of this program. The GUI is created here using NetBeans and has various buttons for user input and information output.
  + - My method for the GUI creation was to use the design tab and place all buttons and fields within the jFrame
  + Buttons solutions:
    - Type of vehicle buttons - sets a variable named selection to a number corresponding with the selection to create the proper type of automobile.
      * When one button is selected it clears the fields not needed for it and deselects the other buttons
      * I got this to mostly work except the electric button only greys out MPG until user types in the weight field. However, the rest of them work well. I did this because I thought something with other fields not being cleared was causing issues when switching vehicle types, then realized that it was not. Kept in, because I thought it’d be okay to have a little extra and am mostly happy with it and prevents two vehicle types from being selected.
    - Clear fields button – clears all button selection and clears all text fields
    - Compute sales tax button – uses n mod 5 to allow array of objects to keep accepting new objects without going out of bounds. N is a variable that increments each time object is created and the mod 5 causes it to drop the extra elements while keeping the newest ones. This also means that later I need to start printing from n%5 location to print in order. Unfortunately, I know how to use mod, but my wording for exactly how it works is sub-par. This part of the code also has if statements to see what the selection variable is to create the proper vehicle and pass in the correct arguments.
    - Display report button – Here the program either loops through the array of objects starting at n%5 and then prints the objects Infront of it to keep it in order, or if the array of objects is than or equal to 5 will print in order normally.
  + Text field solutions
    - Here we needed to have the text fields only accept integers. I achieved this by allowing editable text fields if user entered characters are between 0 and 9 or equal to the Unicode for backspace. I needed the backspace allowed here, because if not my message window will pop up every time backspace is pressed. A popup window will display here if something other than numbers are entered.
  + Note: forgot to declare a few variables at end of project2.java private, did so in final submission java file and works the same.

**Test cases.**

**Test case 1:** create four cars of different types to ensure it will work with less than five objects

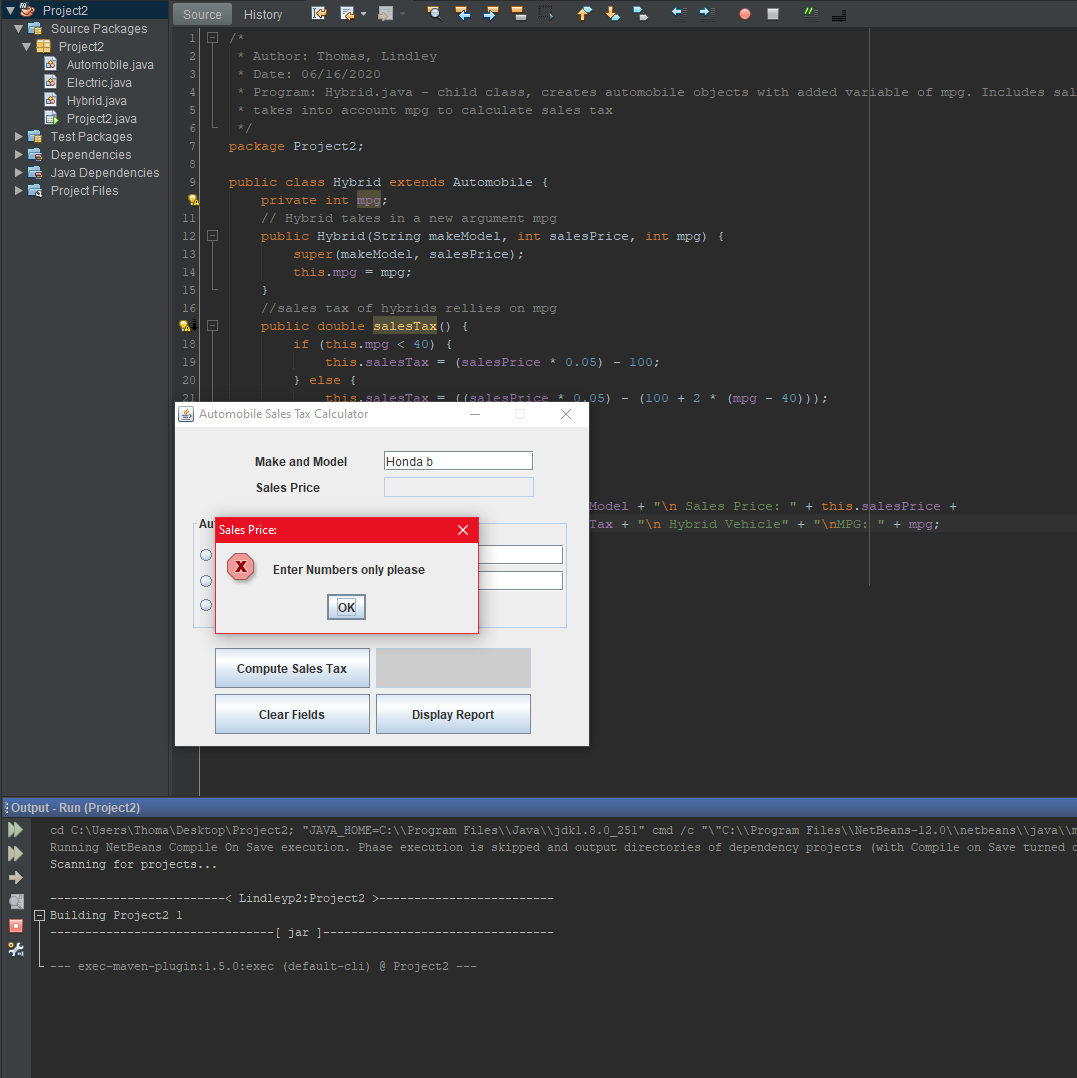
|  |  |  |  |
| --- | --- | --- | --- |
| **Input** | **Expected output** | **Output** | **Pass or Fail** |
| **Honda a**  **50000**  **Hybrid**  **78mpg**  **Toyota a**  **30000**  **Other**  **Tesla c**  **40000**  **Electric**  **2000lbs**  **Ferrari c**  **100000**  **Other** | **Sales tax 2324.0**  Sales tax 1500.0  Sales tax 1800.0  Sales tax 5000.0 | **Make and Model: Honda a**  **Sales Price: 50000**  **Sales Tax: 2324.0**  **Hybrid Vehicle**  **MPG: 78**  **Make and Model: Toyota a**  **Sales Price: 30000**  **Sales Tax: 1500.0**  **Make and Model: Tesla**  **Sales Price: 40000**  **Sales Tax: 1800.0**  **Electric Vehicle**  **Weight: 2000**  **Make and Model: Ferrari**  **Sales Price: 100000**  **Sales Tax: 5000.0** | **Pass** |

****

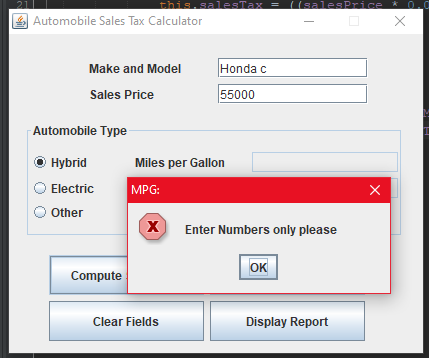
**Test case 2:** eight vehicles of different types, user tries to enter non digit in each text field that only accepts digits, then corrects entry (screen shot of all three below table).

|  |  |  |  |
| --- | --- | --- | --- |
| **Input** | **Expected output** | **Output** | **Pass or Fail** |
| **Honda b**  **60000(tries to enter g)**  **Hybrid**  **38mpg**  **Toyota b**  **60000**  **Other**  **Tesla b**  **40000**  **Electric**  **4000lbs**  **Ferrari b**  **150000**  **Other**  **Honda c**  **55000**  **Hybrid**  **56mpg(tries to enter a)**  **Toyota c**  **70000**  **Other**  **Tesla c**  **80000**  **Electric**  **2000lbs(tries to enter c)**  **Ferrari c**  **100000**  **Other** | Pop-up window  Sales tax 2900.0  Sales tax 3000.0  Sales tax 2850.0  (will only display below)  Sales tax 7500.0  Pop-up window  Sales tax 2618.0  Sales tax 3500.0  Pop-up window  Sales tax 3800.0  Sales tax 5000.0 | **Make and Model: Ferrari b**  **Sales Price: 150000**  **Sales Tax: 7500.0**  **Make and Model: Honda c**  **Sales Price: 55000**  **Sales Tax: 2618.0**  **Hybrid Vehicle**  **MPG: 56**  **Make and Model: Toyota c**  **Sales Price: 70000**  **Sales Tax: 3500.0**  **Make and Model: Tesla c**  **Sales Price: 80000**  **Sales Tax: 3800.0**  **Electric Vehicle**  **Weight: 2000**  **Make and Model: Ferrari c**  **Sales Price: 100000**  **Sales Tax: 5000.0** | **Pass** |

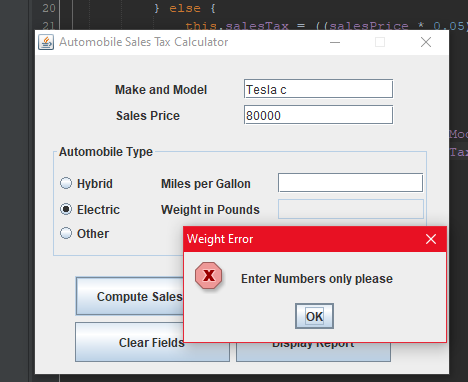
**Popup from ‘g’ entered into sales price (after typing part of the number) for Honda b:**

****

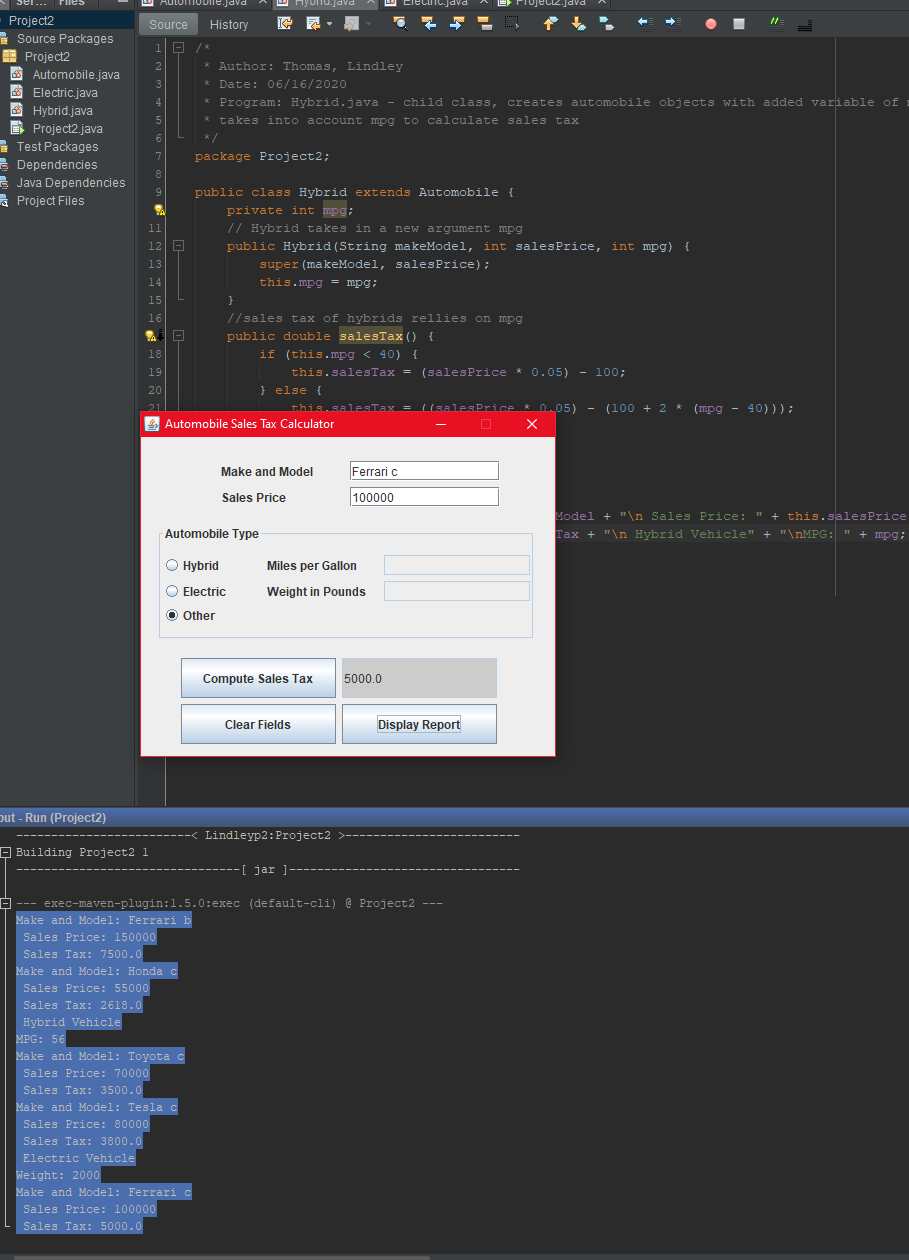
**Popup from ‘a’ entered into MPG (after typing 5) for Honda c:**

****

**Popup from ‘c’ entered into weight (after typing 200) for Tesla c:**



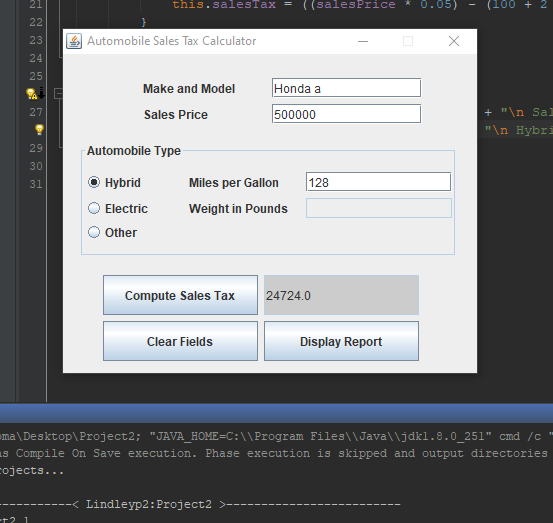
**Screenshot finished:**

****

|  |  |  |  |
| --- | --- | --- | --- |
| **Input** | **Expected output** | **Output** | **Pass or Fail** |
| **Honda a**  **500000**  **Hybrid**  **128mpg**  **Tesla c**  **400000**  **Electric**  **500lbs**  **Ferrari c**  **1000000**  **Hybrid**  **300mpg**  **Tesla d**  **4000000**  **Electric**  **500lbs**  **Ferrari d**  **1000000**  **Hybrid**  **500mpg** | **Sales tax 24724.0**  **Sales tax 19800.0**  **Sales tax 49380.0**  **Sales tax 199800.0**  **Sales tax 198980.0** | **Make and Model: Honda a**  **Sales Price: 500000**  **Sales Tax: 24724.0**  **Hybrid Vehicle**  **MPG: 128**  **Make and Model: Tesla c**  **Sales Price: 400000**  **Sales Tax: 19800.0**  **Electric Vehicle**  **Weight: 500**  **Make and Model: Ferrari c**  **Sales Price: 1000000**  **Sales Tax: 49380.0**  **Hybrid Vehicle**  **MPG: 300**  **Make and Model: Tesla d**  **Sales Price: 4000000**  **Sales Tax: 199800.0**  **Electric Vehicle**  **Weight: 500**  **Make and Model: Ferrari d**  **Sales Price: 4000000**  **Sales Tax: 198980.0**  **Hybrid Vehicle**  **MPG: 500** | **Pass** |

**Test case 3: create 5 vehicles that are ultra-fuel efficient or very light weight, but also very expensive.**

**Screenshots of each in order:**

****

