Here is a detailed walkthough of my Program and its testing methods.

The main entry point of the program starts in the main method of the PayableTest class. Here, I’m initializing 5 different types of Payable objects using the declaration interfaceObj. The initializations pass some hard-coded values into the overloaded constructor of each class object.

A screenshot of a computer

Description automatically generated with medium confidence

The first object is an Invoice type and is initialized by calling on its overloaded constructor, which then calls each of the 4 class seen below in order to initialize its members with the passed values.

A screenshot of a computer

Description automatically generated

A similar process occurs for the initialization of the next objects.

SalariedEmployee is set up to catch an exception if its weeklySalary member is initialized as 0.0 or less.

Text

Description automatically generated

Similarly, HourlyEmployee catches an exception if its wage and hours are invalid, and CommisionPay as well as BasePlusCommission do so if grossSales, commissionRate, or basePay are invalid.

Once all objects are initialized and no exceptions are caught, the objects should be populated with their appropriate values and class type.

Finally, my program will go through a loop to print details for each object.

Graphical user interface, text

Description automatically generated

Output:

A screenshot of a computer

Description automatically generated with medium confidence

Lets also look at the exception handling of the program. To test it out I will initialize an HourlyEmployee object and pass 200 as the value for its hours member.

Text

Description automatically generated