Java Project: Restaurant

Java Concepts Covered:

- Classes and Objects
- Inheritance
- Method Overriding
- ArrayLists
- Control Structures
- Exception
- User Input
- Type Casting

The project allows the user to be able to manage a simplified restaurant menu, including the addition of new menu items.

```
import java.util.ArrayList;

import java.util.InputMismatchException;

import java.util.Scanner;


// Parent class to represent a general menu item

class MenuItem {

   protected String name;

   protected double price;


   public MenuItem(String name, double price) {

      this.name = name;

      this.price = price;

   }


   // Method to display information about the menu item

   public void display() {

      System.out.println("Name: " + name + ", Price: " + price);

   }

}
```

```
// Child class to represent a specialized type of menu item, a Beverage
class Beverage extends MenuItem {
    private boolean isHot;

    public Beverage(String name, double price, boolean isHot) {
        super(name, price);
        this.isHot = isHot;
    }

    // Overloaded method to display information about the beverage
    @Override
    public void display() {
        super.display();
        System.out.println("Is Hot: " + isHot);
    }
}

// Main class to simulate restaurant operations
public class RestaurantApp {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        ArrayList<MenuItem> menu = new ArrayList<>();

        // Add initial items to the menu
        menu.add(new MenuItem("Burger", 5.99));
        menu.add(new MenuItem("Pizza", 8.99));
        menu.add(new Beverage("Coffee", 2.99, true));
        menu.add(new Beverage("Iced Tea", 2.49, false));
```

```
// Main loop for the restaurant app
while (true) {
    try {
        System.out.println("1. Display Menu");
        System.out.println("2. Add Item to Menu");
        System.out.println("3. Exit");
        System.out.print("Enter your choice: ");

        int choice = scanner.nextInt();

        switch (choice) {
            case 1:
                // Display the menu
                System.out.println("Menu:");
                for (MenuItem item : menu) {
                    item.display();
                }
                break;
            case 2:
                // Add a new item to the menu
                System.out.println("1. Regular Item");
                System.out.println("2. Beverage");
                System.out.print("Enter the type of item: ");
                int type = scanner.nextInt();
                scanner.nextLine(); // Clear the buffer

                System.out.print("Enter item name: ");
                String name = scanner.nextLine();
                System.out.print("Enter item price: ");
                double price = scanner.nextDouble();
```

```
                    if (type == 1) {
                        menu.add(new MenuItem(name, price));
                    } else if (type == 2) {
                        System.out.print("Is it a hot beverage? (true/false): ");
                        boolean isHot = scanner.nextBoolean();
                        menu.add(new Beverage(name, price, isHot));
                    } else {
                        System.out.println("Invalid type. Item not added.");
                    }
                    break;
                case 3:
                    System.out.println("Exiting...");
                    return;
                default:
                    System.out.println("Invalid choice. Please try again.");
            }
        } catch (InputMismatchException e) {
            System.out.println("Invalid input. Please enter the correct type.");
            scanner.next(); // Clear the incorrect input
        }
    }
  }
}
```