```
import java.util.*;

class Employee {
   int id;
   String name;
   double salary;

   Employee(int id, String name, double salary) {
      this.id = id;
      this.name = name;
      this.salary = salary;
   }

   public String toString() {
      return "ID: " + id + ", Name: " + name + ", Salary: " + salary;
   }
}

public class PayrollSystem {
   static Stack<Employee> history = new Stack<>();
   static List<Employee> employees = new ArrayList<>();

   public static void addEmployee(int id, String name, double salary) {
      try {
         if (salary < 0) throw new IllegalArgumentException("Salary cannot be negative!");
         Employee emp = new Employee(id, name, salary);
         employees.add(emp);
         history.push(emp);
         System.out.println("Employee added: " + emp);
      } catch (Exception e) {
         System.out.println("Error: " + e.getMessage());
      }
   }

   public static void displayEmployees() {
      if (employees.isEmpty()) {
         System.out.println("No employees in payroll.");
      } else {
         System.out.println("Payroll List:");
         for (Employee emp : employees) {
            System.out.println(emp);
         }
      }
   }

   public static void undoLast() {
      try {
         if (history.isEmpty()) throw new EmptyStackException();
         Employee removed = history.pop();
         employees.remove(removed);
         System.out.println("Undo successful. Removed: " + removed);
```

```
        } catch (EmptyStackException e) {
            System.out.println("No operations to undo!");
        }
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int choice;

        do {
            System.out.println("\n--- Payroll Menu ---");
            System.out.println("1. Add Employee");
            System.out.println("2. Display Employees");
            System.out.println("3. Undo Last Operation");
            System.out.println("4. Exit");
            System.out.print("Enter your choice: ");
            choice = sc.nextInt();

            switch (choice) {
                case 1:
                    System.out.print("Enter Employee ID: ");
                    int id = sc.nextInt();
                    sc.nextLine(); // consume newline
                    System.out.print("Enter Employee Name: ");
                    String name = sc.nextLine();
                    System.out.print("Enter Employee Salary: ");
                    double salary = sc.nextDouble();
                    addEmployee(id, name, salary);
                    break;

                case 2:
                    displayEmployees();
                    break;

                case 3:
                    undoLast();
                    break;

                case 4:
                    System.out.println("Exiting program...");
                    break;

                default:
                    System.out.println("Invalid choice! Try again.");
            }
        } while (choice != 4);

        sc.close();
    }
}
```