Yes! You can format the output of your loan details into a table-like structure using ASCII art. Here's how you can modify the `MauriBankLoanCalculator` to display the loan details in a tabulated form using ASCII characters:

Updated `MauriBankLoanCalculator` with ASCII table:

```java
import java.util.InputMismatchException;
import java.util.Scanner;

// Base Loan class (abstraction and inheritance)
public abstract class Loan {
    protected double loanAmount;
    protected int loanTerm; // in years
    protected double interestRate;

    public Loan(double loanAmount, int loanTerm) {
        this.loanAmount = loanAmount;
        this.loanTerm = loanTerm;
        setInterestRate(); // Interest rate is set by the bank for each loan type
    }

    // Set interest rate - must be implemented by subclasses
    protected abstract void setInterestRate();

    // Method to calculate monthly payment
    public double calculateMonthlyPayment() {
        int totalMonths = loanTerm * 12;
        double monthlyInterestRate = (interestRate / 100) / 12;
        return (loanAmount * monthlyInterestRate * Math.pow(1 + monthlyInterestRate, total
                (Math.pow(1 + monthlyInterestRate, totalMonths) - 1);
    }

    // Getter for loan type name - must be implemented by subclasses
    public abstract String getLoanType();

    // Getter for interest rate to display it to the user
    public double getInterestRate() {
        return interestRate;
    }
}

// Home Loan class (inheritance and polymorphism)
class HomeLoan extends Loan {
    public HomeLoan(double loanAmount, int loanTerm) {
        super(loanAmount, loanTerm);
```

```
    }

    @Override
    protected void setInterestRate() {
        this.interestRate = 5.0; // Bank-defined interest rate for home loan
    }

    @Override
    public String getLoanType() {
        return "Home Loan";
    }
}

// Car Loan class (inheritance and polymorphism)
class CarLoan extends Loan {
    public CarLoan(double loanAmount, int loanTerm) {
        super(loanAmount, loanTerm);
    }

    @Override
    protected void setInterestRate() {
        this.interestRate = 6.5; // Bank-defined interest rate for car loan
    }

    @Override
    public String getLoanType() {
        return "Car Loan";
    }
}

// Personal Loan class (inheritance and polymorphism)
class PersonalLoan extends Loan {
    public PersonalLoan(double loanAmount, int loanTerm) {
        super(loanAmount, loanTerm);
    }

    @Override
    protected void setInterestRate() {
        this.interestRate = 8.0; // Bank-defined interest rate for personal loan
```

```
    }

    @Override
    public String getLoanType() {
        return "Personal Loan";
    }
}

// Main application class
public class MauriBankLoanCalculator {

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

        try {
            // User inputs
            System.out.println("Welcome to MauriBank Loan Department!");

            System.out.print("Enter your name: ");
            String userName = scanner.nextLine();

            System.out.print("Enter your age: ");
            int age = scanner.nextInt();

            // Check if user age is valid for loan eligibility
            if (age < 18 || age > 65) {
                throw new IllegalArgumentException("You must be between 18 and 65 years ol
            }

            // Calculate maximum loan term based on age (up to 65 years)
            int maxLoanTerm = 65 - age;

            System.out.println("Hello " + userName + ", please choose a loan type:");
            System.out.println("1. Home Loan");
            System.out.println("2. Car Loan");
            System.out.println("3. Personal Loan");

            int loanChoice = scanner.nextInt();
```

```
        if (loanChoice < 1 || loanChoice > 3) {
            throw new IllegalArgumentException("Invalid loan type selected. Please cho
        }

        System.out.print("Enter the loan amount you wish to take: ");
        double loanAmount = scanner.nextDouble();

        if (loanAmount <= 0) {
            throw new IllegalArgumentException("Loan amount must be greater than zero.
        }

        System.out.printf("Enter the loan term in years (maximum %d years): ", maxLoan
        int loanTerm = scanner.nextInt();

        // Validate the loan term entered by the user
        if (loanTerm <= 0 || loanTerm > maxLoanTerm) {
            throw new IllegalArgumentException(
                    String.format("Invalid loan term. You can only repay your loan ove
        }

        Loan loan = null;

        // Determine loan type based on user's choice
        switch (loanChoice) {
            case 1:
                loan = new HomeLoan(loanAmount, loanTerm);
                break;
            case 2:
                loan = new CarLoan(loanAmount, loanTerm);
                break;
            case 3:
                loan = new PersonalLoan(loanAmount, loanTerm);
                break;
        }

        // Display loan details including interest rate and calculate monthly payment
        double monthlyPayment = loan.calculateMonthlyPayment();

        // ASCII Art Table to Display Loan Details
```

```
            System.out.println("+----------------+----------------------+");
            System.out.println("| Loan Detail    | Value                |");
            System.out.println("+----------------+----------------------+");
            System.out.printf("| Loan Type      | %-20s |\n", loan.getLoanType());
            System.out.println("+----------------+----------------------+");
            System.out.printf("| Loan Amount    | %.2f               |\n", loanAmount);
            System.out.println("+----------------+----------------------+");
            System.out.printf("| Loan Term      | %d years             |\n", loanTerm);
            System.out.println("+----------------+----------------------+");
            System.out.printf("| Interest Rate  | %.2f%%               |\n", loan.getInte
            System.out.println("+----------------+----------------------+");
            System.out.printf("| Monthly Payment| %.2f               |\n", monthlyPayment)
            System.out.println("+----------------+----------------------+");

        } catch (InputMismatchException e) {
            System.out.println("Invalid input. Please enter the correct data type.");
        } catch (IllegalArgumentException e) {
            System.out.println(e.getMessage());
        } finally {
            scanner.close();
        }
    }
}
```

What's new:

- The code now includes an **ASCII Art Table** to neatly display the loan details such as loan type, amount, term, interest rate, and monthly payment.
- Each field is formatted within the table, providing a clear, tabulated summary of the loan's key details.

Example Output:

```
+----------------+----------------------+
| Loan Detail    | Value                |
+----------------+----------------------+
| Loan Type      | Home Loan            |
+----------------+----------------------+
| Loan Amount    | 200000.00            |
+----------------+----------------------+
| Loan Term      | 20 years             |
+----------------+----------------------+
| Interest Rate  | 5.00%                |
+----------------+----------------------+
| Monthly Payment| 1319.91              |
+----------------+----------------------+
```

This table enhances the readability of the output and gives the user a clear understanding of their loan details in a well-structured format.