北京邮电大学软件学院

实验报告

（注意：此处填写实验指导手册中的实验名称），提交时将这个文本框删除

**提交时，将本注释框删除**

课程名称： Java SE 程 序 设 计

项目名称：

项目完成人：

姓名：

学号：

指导教师： 程保中 孙艺

日 期： 年 月 日

（正文文字，请使用【正文缩进】样式）

（标题编号请使用自动编号。在样式表中选中标题级别样式即可自动编号。）

# 实验概述

## 实验目的

（说明通过本实验希望达到的目的。要根据你自己的理解撰写。例如，学习使用某种工具、学习某种方法、理解某种概念、锻炼某种实践能力等。一般不超过100个字）

## 实验内容

（说明本实验的内容。以教师所给的实验要求为蓝本，将所需完成的工作细化成5-8条小任务或者内容，按照大致的顺序罗列在下方表格中之。注意，不能照抄教师所给实验要求。本部分内容不要超过200字）

|  |  |
| --- | --- |
|  | 任务说明 |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

## 实验环境

（说明本实验需要的环境）

### 硬件环境

（说明本实验所需的硬件环境，例如计算机型号、配置等）

### 网络环境

（说明本实验所需的网络环境，例如网络设备型号、配置等）

（如果实验中搭建了网络，或者网络配置是实验中的重要部分，请说明整个实验外部网络环境的状况。）

### 软件环境

（说明本实验所用的软件环境。例如操作系统类型、版本，所用的软件名称、版本等。此处无需说明软件如何配置。软件配置应作为实验方案的一部分放到第二章中介绍）

## 实验结果

（简要说明实验完成情况，本部分 不超过100字）

# 实验内容

# Project #N分析设计 （N = 1，2，3，4，5 …）

（附上实验文档，如：问题分析、算法、UML类图、程序、仿真结果、运行结果、调试心得等，具体内容根据实验要求来定。源代码请附在这里。源代码排版请特别注意，用5 号字体，行间距为单倍行距。注意节省空间，不要浪费纸张。）

## N.1问题分析

（说明你对这个问题的理解，包括，这个程序要解决什么问题、功能、性能、健壮性等）

### N.1.1 目标

（说明这个程序要解决什么问题，达到的目标）

### N.1.2功能

（说明这个程序的各种功能，参见下例 N.1.2.1和 N.1.2.2。在撰写正文时，请将下面的例子删除）

#### N.1.2.1功能一：来电振铃

当手机接收到网络的振铃信号时，能够激活手机屏幕，并且通过手机扬声器或者耳机播放预先设定的声音；

当用户按挂机键或者接听键后，停止铃声；

若铃声持续2分钟后，用户没有按挂机键或者接听键，则向电话网络反馈信号“无人”

#### N.1.2.2功能二：查找联系人

当用户按“查找”按钮之后，程序显示输入框，等待用户输入。用户输入文字的同时，对用户的通信录按照字母或者汉语拼音进行检索

### N.1.3性能

（说明这个程序主要功能的性能参数。在撰写正文时，请删除例子）

#### N.1.3.1查找联系人的性能

在用户输入文字的同时，程序在每次输入后的0.4秒之内响应并将结果显示到屏幕上

#### N.1.3.2健壮性

（说明这个程序应对错误输入、异常情况的能力。在撰写正文时，请删除例子）

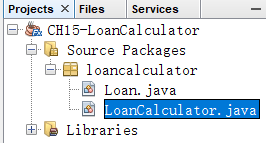
#### N.1.3.3异常输入处理

当用户在检索联系人时，输入的字符超过20个字符（字符可以是单字节语言的字符，也可以是双字节语言的字符），则不再接受用户的输入

## N.2设计方案

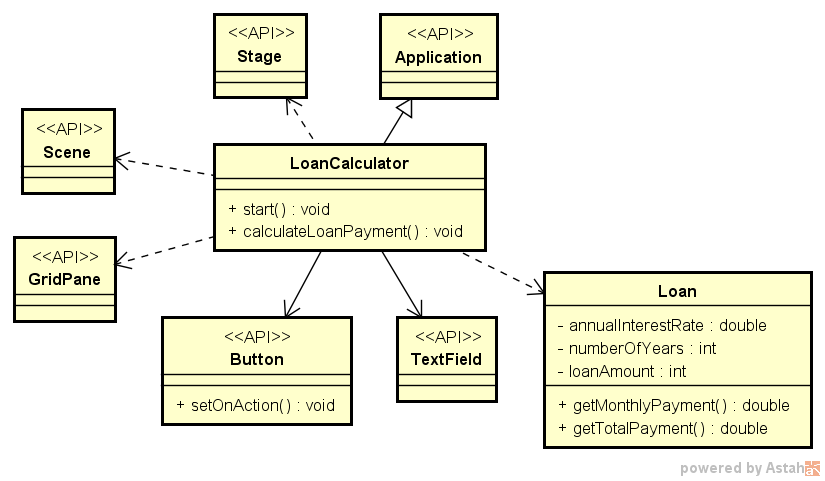
（在此处，用文字描述整体的设计，包括，分成多少个模块（Java package），多少个类。每个模块、类负责的大致功能； 此外，还需要包含一个程序整体的流程图，反映各个模块和类之间的关系；但是避免绘制过于详细的流程）

### N.2.1模块划分



本项目只包含一个模块 loancalculator

### N.2.2类图



注：其中API类是可选项。程序员自己定义的类（LoanCalaculator，Loan）是必选项。

### N.2.3算法

（在此处，用流程图描绘重要的算法，例如排序、查找等。一般的程序代码不必在此说明）

#### N.2.1.1算法一：xxxxxx

#### N.2.1.2算法二：xxxxxx

## N.3实验结果

（说明实验完成情况，有需要文字回答的题目，以及实验结果截图，请写在这里）

（在此处，详细描述运行的结果。包括：不同输入对应的不同结果，异常情况的执行结果）

## N.4 调试心得

（总结你在调试程序时的收获，对于某一类或者某几类警告、错误的处理方法。这些心得，能够对学习Java语言程序设计的新手具有一些指导作用）

## N.5源程序

/\* 这是源代码的排版样式，底纹是灰色的，字体使用Consolas 5号字

\* 如果没有Consolas字体，则切换为 courier new 字体

\* 源代码务必注意缩进

\* 每个程序文件之前，必须用注释说明该文件的名字

\*/

public class Loan {

private double annualInterestRate;

private int numberOfYears;

private double loanAmount;

private java.util.Date loanDate;

public Loan() {

this(2.5, 1, 1000);

}

public Loan(double annualInterestRate, int numberOfYears,

double loanAmount) {

this.annualInterestRate = annualInterestRate;

this.numberOfYears = numberOfYears;

this.loanAmount = loanAmount;

loanDate = new java.util.Date();

}

public double getAnnualInterestRate() {

return annualInterestRate;

}

public void setAnnualInterestRate(double annualInterestRate) {

this.annualInterestRate = annualInterestRate;

}

public int getNumberOfYears() {

return numberOfYears;

}

public void setNumberOfYears(int numberOfYears) {

this.numberOfYears = numberOfYears;

}

public double getLoanAmount() {

return loanAmount;

}

public void setLoanAmount(double loanAmount) {

this.loanAmount = loanAmount;

}

public double getMonthlyPayment() {

double monthlyInterestRate = annualInterestRate / 1200;

double monthlyPayment = loanAmount \* monthlyInterestRate / (1

- (1 / Math.pow(1 + monthlyInterestRate,

numberOfYears \* 12)));

return monthlyPayment;

}

public double getTotalPayment() {

double totalPayment = getMonthlyPayment() \* numberOfYears \* 12;

return totalPayment;

}

public java.util.Date getLoanDate() {

return loanDate;

}

}

public class LoanCalculator extends Application {

private TextField tfAnnualInterestRate = new TextField();

private TextField tfNumberOfYears = new TextField();

private TextField tfLoanAmount = new TextField();

private TextField tfMonthlyPayment = new TextField();

private TextField tfTotalPayment = new TextField();

private Button btCalculate = new Button("Calculate");

@Override // Override the start method in the Application class

public void start(Stage primaryStage) {

// Create UI

GridPane gridPane = new GridPane();

gridPane.setHgap(5);

gridPane.setVgap(5);

gridPane.add(new Label("Annual Interest Rate:"), 0, 0);

gridPane.add(tfAnnualInterestRate, 1, 0);

gridPane.add(new Label("Number of Years:"), 0, 1);

gridPane.add(tfNumberOfYears, 1, 1);

gridPane.add(new Label("Loan Amount:"), 0, 2);

gridPane.add(tfLoanAmount, 1, 2);

gridPane.add(new Label("Monthly Payment:"), 0, 3);

gridPane.add(tfMonthlyPayment, 1, 3);

gridPane.add(new Label("Total Payment:"), 0, 4);

gridPane.add(tfTotalPayment, 1, 4);

gridPane.add(btCalculate, 1, 5);

// Set properties for UI

gridPane.setAlignment(Pos.CENTER);

tfAnnualInterestRate.setAlignment(Pos.BOTTOM\_RIGHT);

tfNumberOfYears.setAlignment(Pos.BOTTOM\_RIGHT);

tfLoanAmount.setAlignment(Pos.BOTTOM\_RIGHT);

tfMonthlyPayment.setAlignment(Pos.BOTTOM\_RIGHT);

tfTotalPayment.setAlignment(Pos.BOTTOM\_RIGHT);

tfMonthlyPayment.setEditable(false);

tfTotalPayment.setEditable(false);

GridPane.setHalignment(btCalculate, HPos.RIGHT);

// Process events

btCalculate.setOnAction(e -> calculateLoanPayment());

// Create a scene and place it in the stage

Scene scene = new Scene(gridPane, 400, 250);

primaryStage.setTitle("LoanCalculator"); // Set title

primaryStage.setScene(scene); // Place the scene in the stage

primaryStage.show(); // Display the stage

}

private void calculateLoanPayment() {

// Get values from text fields

double interest =

Double.parseDouble(tfAnnualInterestRate.getText());

int year = Integer.parseInt(tfNumberOfYears.getText());

double loanAmount = Double.parseDouble(tfLoanAmount.getText());

// Create a loan object. Loan defined in Listing 10.2

Loan loan = new Loan(interest, year, loanAmount);

// Display monthly payment and total payment

tfMonthlyPayment.setText(String.format("$%.2f",

loan.getMonthlyPayment()));

tfTotalPayment.setText(String.format("$%.2f",

loan.getTotalPayment()));

}

}