

实验二 利率计算器

1. 程序介绍

该程序为名称为利率计算器，设计的思路为仿照招商银行在线利率计算器，用 Qt 框架配合 c++ 语言来完成。这里实现了等额本息和等额本金两种贷款模式，只需要输入贷款年限，贷款金额，贷款利率就可以进行计算出月均还款和利息总额。为了更好的呈现一个美好的 UI 界面，在保证计算器严谨的同时，添加彩色触显，来增强用户体验感。

2. 操作说明

2.1 等额本息

等额本息是指一种贷款的还款方式，指在还款期内，每月偿还同等数额的贷款(包括本金和利息)。这里选择等额本息，以 20 年贷款 20 万.利率 5% 计算，结果如图 1 所示

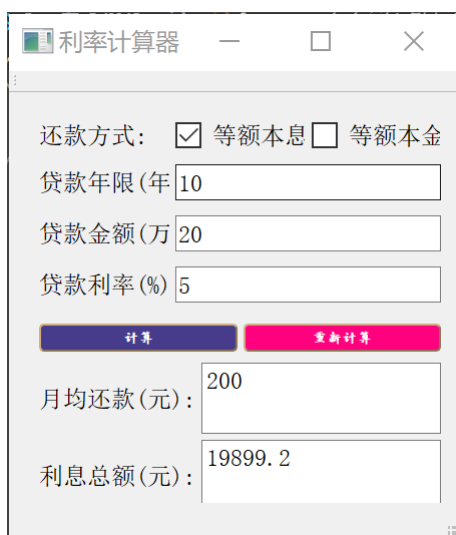


图 1 等额本息展示

2.2 等额本金

等额本金是指一种贷款的还款方式，是在还款期内把贷款数总额等分，每月偿还同等数额的本金和剩余贷款在该月所产生的利息，这样由于每月的还款本金

额固定，而利息越来越少，借款人起初还款压力较大，但是随时间的推移每月还款数也越来越少。这里选择等额本金，以 20 年贷款 20 万.利率 5%计算，结果如图 2 所示

利率计算器

还款方式: ☐ 等额本息 ☒ 等额本金

贷款年限(年)

贷款金额(万)

贷款利率(%)

月均还款(元):

利息总额(元):

图 2 等额本金展示

3.设计理念

3.1 设计目标

该项目是想设计一个利率计算的程序,这里实现了等额本息和等额本金两种贷款模式,分别对应不同的需求,在程序中只需要输入贷款年限,贷款金额,贷款利率就可以进行计算出月均还款和利息总额。其运行界面如图 3 所示。

利率计算器

还款方式: ☐ 等额本息 ☐ 等额本金

贷款年限(年)

贷款金额(万)

贷款利率(%)

月均还款(元):

利息总额(元):

图 3 利率计算器页面

3.2 设计分析和算法分析

3.2.1 页面搭建

这里采用 Qt 框架自带的 UI 界面进行美化排布，效果如下图所示，利用 UI 自动生成按钮等类，在页面美化中，添加样本样式，利用 Qt 自带的 QSS 进行美化（操作如前端 CSS 类似）。



图 4UI 界面美化

3.2.2 按钮绑定

利用 UI 生成的类，添加槽函数进行关联绑定，生成类如图 13 所示，在 mainwindow.cpp 析构函数中进行绑定操作。代码如下：

对象	类
▼ MainWindow	QMainWindow
▼ centralWidget	QWidget
▼ gridLayout	QGrid...you
checkbox	QCheckBox
checkbox_2	QCheckBox
label	QLabel
label_2	QLabel
label_3	QLabel
label_4	QLabel
lineEdit	QLineEdit
lineEdit_2	QLineEdit
lineEdit_3	QLineEdit
▼ horizontalLayout	QHor...yo
pushButton	QPushButton
pushButton_2	QPushButton
▼ gridLayout_2	QGrid...you
label_5	QLabel
label_6	QLabel
label_7	QLabel
textBrowser_4	QTextBrowser
textBrowser_5	QTextBrowser
textBrowser_6	QTextBrowser
menuBar	QMenuBar
mainToolBar	QToolBar
statusBar	QStatusBar

图 5 按钮类

2.2.3 模式判断算法

这里有两钟计算利率的方法。只能勾选一种，在这里我们利用一个全局变量来控制计算模式如何，对于还款方式进行构建，代码如下：

```
//等额本息
void MainWindow::on_checkBox_clicked()
{
    if(temp_1==1) temp_1=0;
    else temp_1=1;
    if(temp_2==2&&err == 0)
    {
        QMessageBox::critical(NULL, "错误信息", "请选择一种还款方式", QMessageBox::Yes |
        QMessageBox::No, QMessageBox::Yes);
        ui->textBrowser_6->clear();
    }
}
```

```
        ui->textBrowser_5->clear();
        ui->textBrowser_4->clear();
        ui->lineEdit->clear();
        ui->lineEdit_2->clear();
        ui->lineEdit_3->clear();
        ui->checkBox_2->click();

        //err = 1;
    }
}

//等额本金
void MainWindow::on_checkBox_2_clicked()
{
    if(temp_2==2) temp_2=0;
    else temp_2=2;
    if(temp_1==1)
    {
        QMessageBox::critical(NULL, "错误信息", "请选择一种还款方式", QMessageBox::Yes |
QMessageBox::No, QMessageBox::Yes);
        ui->textBrowser_6->clear();
        ui->textBrowser_5->clear();
        ui->textBrowser_4->clear();
        ui->lineEdit->clear();
        ui->lineEdit_2->clear();
        ui->lineEdit_3->clear();

        ui->checkBox->click();
    }
}
```

代码块 1 模式判断代码

2.2.4 重新计算算法

在计算后考虑到用户需要清空，这里重新计算提供一个清空操作，将 UI 中 lineEdit 类都置空，并模式初始化为 0.代码如下：

```
void MainWindow::on_pushButton_2_clicked(bool checked)
{
    ui->textBrowser_6->clear();
    ui->textBrowser_5->clear();
    ui->textBrowser_4->clear();
```

```
ui->lineEdit->clear();
ui->lineEdit_2->clear();
ui->lineEdit_3->clear();
temp_1=0;
temp_2=0;
}
```

代码块 2 重新计算代码

2.2.5 计算算法

这里首先要明白计算原理

1. 等额本息

等额本息是指一种贷款的还款方式，指在还款期内，每月偿还同等数额的贷款(包括本金和利息)。等额本息还款法即借款人每月按相等的金额偿还贷款本息，其中每月贷款利息按月初剩余贷款本金计算并逐月结清。

计算公式如下：

$$\text{每月还款额} = [\text{贷款本金} \times \text{月利率} \times (1 + \text{月利率})^{\text{还款月数}}] \div [(1 + \text{月利率})^{\text{还款月数}} - 1]$$

2. 等额本金

等额本金是指一种贷款的还款方式，是在还款期内把贷款数总额等分，每月偿还同等数额的本金和剩余贷款在该月所产生的利息，这样由于每月的还款本金额固定，而利息越来越少，借款人起初还款压力较大，但是随时间的推移每月还款数也越来越少。

计算公式如下：

$$\text{每月还款金额} = (\text{贷款本金} / \text{还款月数}) + (\text{本金} - \text{已归还本金累计额}) \times \text{每月利率}$$

在计算的基础上加入报错功能，提醒功能，增强人机交互感，代码如下：

```
void MainWindow::on_pushButton_clicked()
{
    double num1=ui->lineEdit->text().toInt();    //贷款年限（年）
    double num2=ui->lineEdit_2->text().toInt();    //贷款金额（万元）
    double num3=ui->lineEdit_3->text().toInt()/100.0;    //贷款利率
    double yue_num3=num3/12.0;    //月利率
    double yue_num2=num2/12.0;    //
```

```
if(temp_1==0&&temp_2==0)
{
    QMessageBox::critical(NULL, "错误信息", "未选择还款方式", QMessageBox::Yes |
QMessageBox::No, QMessageBox::Yes);
    //err = 1;
}
else if(temp_1==1&&temp_2==2)
{
    QMessageBox::critical(NULL, "错误信息", "请选择一种还款方式", QMessageBox::Yes |
QMessageBox::No, QMessageBox::Yes);
    //err = 1;
}
else if(num1<=0)
    QMessageBox::critical(NULL, "错误信息", "未输入有效贷款年限信息",
QMessageBox::Yes | QMessageBox::No, QMessageBox::Yes);
else if(num2<=0)
    QMessageBox::critical(NULL, "错误信息", "未输入有效贷款金额信息",
QMessageBox::Yes | QMessageBox::No, QMessageBox::Yes);
else if(num3<=0)
    QMessageBox::critical(NULL, "错误信息", "未输入有效贷款利率信息",
QMessageBox::Yes | QMessageBox::No, QMessageBox::Yes);
double ans=num1*num2*num3;
if(num1<=0 || num2<=0 || num3<=0 || (temp_1==0&&temp_2==0) || (temp_1==1&&temp_2==2))
{
    ui->textBrowser_6->clear();
    ui->textBrowser_5->clear();
    ui->textBrowser_4->clear();
    ui->lineEdit->clear();
    ui->lineEdit_2->clear();
    ui->lineEdit_3->clear();
    return;
}
if(temp_1==1)
{
    double nn=-((num2*num3*((1+yue_num3)*num1*12-(1+yue_num3)*(num1*12-
1)))/((1+yue_num3)*num1*12-1))*12-num2);
    ui->textBrowser_6->setText(QString::number((nn/10.0+num2)*10000)); //还款总额
    ui->textBrowser_5->setText(QString::number(nn*1000)); //利息总额

    ui->textBrowser_4->setText(QString::number((num2*yue_num3*(1+yue_num3)*num1*12.0)/(1+yue_
num3)*num1*12.0-1000)); //月均还款
}
```

```
if(temp_2==2)
{
    double yue_huan=(num2/num1*12)+(num2)*yue_num3;
    double yue_lixi=(num2)*yue_num3;
    double
total_lixi=((num2/num1*12+num2*yue_num3)+num2/num1*12*(1+yue_num3))/2*num1*12-num2;
    ui->textBrowser_6->setText(QString::number(((total_lixi/10000+num2)*10000)));
//还款总额
    ui->textBrowser_5->setText(QString::number(total_lixi)); //利息总额
    ui->textBrowser_4->setText(QString::number(yue_huan*100));
}
}
```

代码块 3 计算原理代码

3.3 类图关系

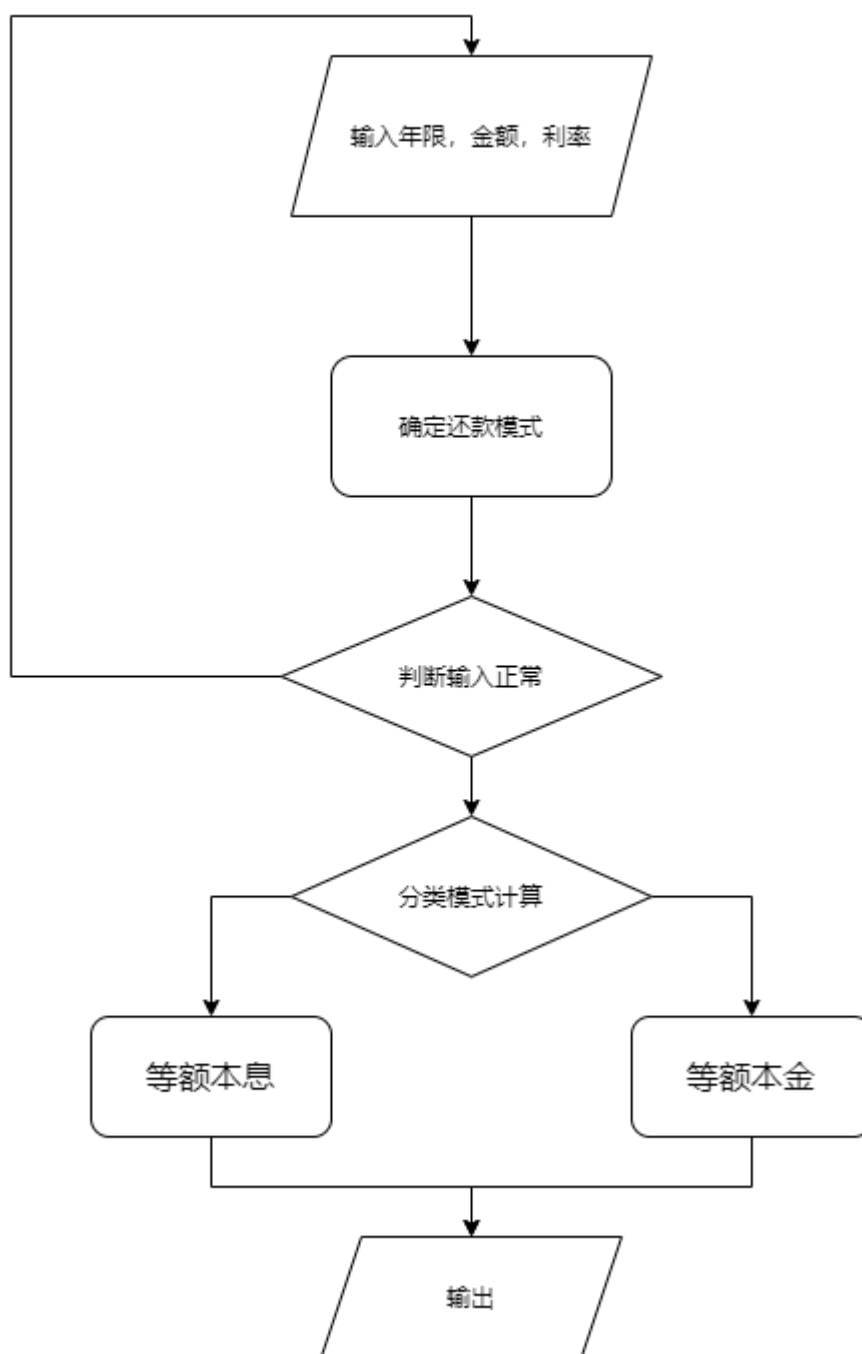


图 6 利率计算流程图

4.程序展示

一些操作实际展示，展示效果如下图 7：



图 7 成果展示图

5.总结思考

通过使用 Qt 应用框架实现了人机交互界面的利率计算器,采用 Qt 信号槽机制实现利率的运算并对可能出现的错误进行预判,并反馈警告。该程序是一个集继承、图形界面、事件处理等面向对象编程知识的综合应用的实例程序。