

# Python大作业

---

姓名：孔渊媛

学号：5141209033

学院：安泰经济与管理学院

日期：2015年12月24日

# Contents

简介 .....	3
简易使用教程 .....	4
实现方法 .....	6
不足与补充 .....	10

# 简介

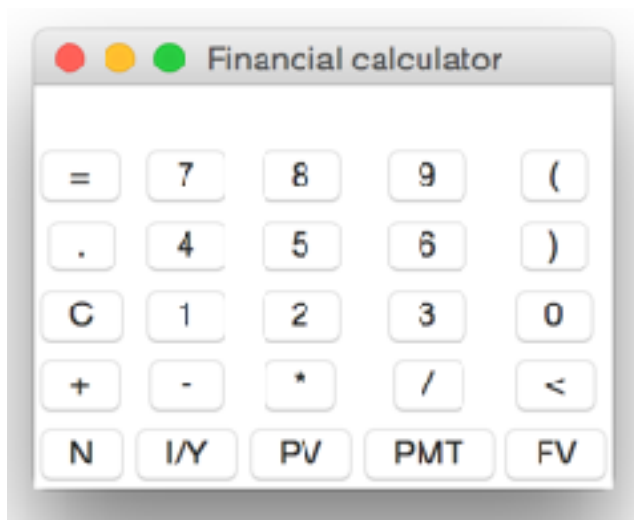
本次大作业，我结合自己所学专业——经济类金融方向设计并实现了一个简易版金融计算器程序。

金融计算器广泛应用于银行、证券、保险、房地产、投资以及商业等相关行业，专业版金融计算器可以方便解决有关货币的时间价值和分期付款、非均匀现金流和分组现金流、债券、折旧、统计回归、利率转换、盈亏平衡等问题。可以免去普通计算器繁琐的函数输入和问题求解，具有很强的针对性和实用性。

我设计的简易版金融计算器可以实现以下功能：

1. 一般四则运算
2. 清除计算器输入项和存储器
3. 货币的时间价值和分期付款工作表
  - 3.1 计算基本贷款偿清年数
  - 3.2 计算基本贷款的每期付款额
  - 3.3 计算年金的现值
  - 3.4 计算年金的终值

# 简易使用教程



## 2. 清除计算器输入项和存储器

按键：C

### 3.1 计算基本贷款偿清年数

某笔抵押贷款,总额\$75,000,年付款额\$5100,该笔贷款的年利率为5.5%,求多少年偿清贷款?

操作	按键
输入年利率	0.055 I/Y
输入贷款金额	75000 PV
每期付款额	5100 PMT
计算付款期数	N

### 3.2 计算基本贷款的每期付款额

例：计算基本贷款的每期付款额（年付款额）,该抵押贷款期限为30年,年利率为5.5%,抵押金额为\$75,000。

操作	按键
使用付款乘子输入付款期数	30 N
输入年利率	0.055 I/Y
输入贷款金额	75000 PV
计算每期付款额	PMT

### 3.3 计算年金的现值

F公司购买一台新设备,预计使用十年,每年可节约费用\$20,000。假设年贴现利率为10%,按普通年金计算所节约费用的现值为多少?

操作	按键
使用付款乘子输入付款期数	10 N
输入年利率	0.1 I/Y
输入付款额	20000 PMT
计算年金现值	PV

### 3.4 计算年金的终值

F公司购买一台新设备,预计使用十年,每年可节约费用\$20,000。假设年贴现利率为10%,按普通年金计算所节约费用的终值为多少?

操作	按键
使用付款乘子输入付款期数	10 N
输入年利率	0.1 I/Y
输入付款额	20000 PMT
计算年金现值	FV

# 实现方法

## 一、图形界面(GUI)

1. 使用Python标准库Tkinter进行界面设计，主要涉及button、label构件。

---

```
#Financial calculator

import Tkinter
import math

#intialize
root = Tkinter.Tk()
root.title('Financial calculator')

#label(a display screen)
s = ''
label = Tkinter.Label(root, text = s, fg='blue')
label.grid(row = 0, column = 0, columnspan = 4)
```

2. 使用Grid布局管理器将基本构件摆放 to 指定位置。

例如:

```
#button
for i in range(4):
    bot = Tkinter.Button(text = '+-*/'[i])
    bot.grid(row = 1, column = i)
    bot.bind('<Button-1>', operation)

bot = Tkinter.Button(text = '=', relief='raised')
bot.grid(row = 1, column = 0)
bot.bind('<Button-1>', equal)
```

## 二、功能实现

1. 基本按钮的功能实现

定义event，然后将button和event bind在一起。

例如：clear按键的功能实现

```
def clear(event):
    global s
    label['text'] = s
    s = ''

bot = Tkinter.Button(text = 'C')
bot.grid(row = 3, column = 0)
bot.bind('<Button-1>', clear)
```

## 2. 高级按钮的功能实现

try-except的使用，使得高级按钮能区分是输入还是计算功能。

### (1) N(分期付款的期数)

```
def handleN(event):
    global label, N, I, PV, PMT, FV
    try:
        try:
            q = 1 / (1 + I)
            tmp = -(PV / PMT * (1 - q) - 1)
            res = math.log(tmp) / math.log(q)
            res = int(res*100)/100.00
            label['text'] = str(res)
            N = I = PV = PMT = FV = None
            print res
        except:
            print PMT, I
            res = math.log(FV / PMT * I + 1) / math.log(1 + I)
            label['text'] = str(res)
            N = I = PV = PMT = FV = None
    except:
        equal(event)
        try:
            N = eval(label['text'])
            print N
        except:
            pass
```

如果同时输入了符合条件的I/Y、PV、PMT或者I/Y、FV、PMT，则按下N键即根据公式计算输出N。（年金的次数计算）  
否则直接输出输入的N值。

### (2) I/Y(分期付款的利率)

由于计算公式比较复杂，这里就直接输出输入的I/Y值了。

```
def handleIY(event):
    global label, N, I, PV, PMT, FV
    equal(event)
    try:
        I = eval(label['text'])
        print I
    except:
        pass
```

### (3) PV(年金的现值)

```
def handlePV(event):
    global label, N, I, FV, PMT, PV
    try:
        q = 1 / (1 + I)
        print q, N, PMT
        PV = (1 - q ** (N + 1)) / (1 - q) * PMT
        PV = int(PV*100)/100.00
        label['text'] = str(PV)
        N = I = FV = PMT = PV = None
    except:
        equal(event)
        try:
            PV = eval(label['text'])
        except:
            pass
```

如果同时输入了符合条件的I/Y、N、PMT，则按下PV键即根据公式计算出PV。

否则直接输出输入的PV值。

### (4) FV(年金的未来值)

同PV（除了公式不同）

### (5) PMT（分期付款额）



```

def handlePMT(event):
    global label, N, I, PV, PMT, FV
    try:
        try:
            q = 1 / (1 + I)
            PMT = PV / (1 - q ** (N + 1)) * (1 - q)
            PMT=int(PMT*100)/100.00
            label['text'] = str(PMT)
            N = I = PV = PMT = FV = None
        except:
            PMT = FV * i / (1 + I) ** N
            label['text'] = str(PMT)
            N = I = PV = PMT = FV = None
    except:
        equal(event)
    try:
        PMT = eval(label['text'])
        print PMT
    except:
        pass

```

如果同时输入了符合条件的I/Y、PV、N或者I/Y、FV、N，则按下PMT键即根据公式计算输出PMT。

否则直接输出输入的PMT值。

注：每一次计算完成后都自动清空存储的值。

## 不足与补充

- 1.功能比起完整版的专业金融计算器还少了许多。
- 2.无记忆功能，每次计算完成后都会自动清空存储的值。
- 3.键盘界面还不够美观。