

嵌入式手机终端开发技术

大作业报告

（ 2020 / 2021学年 第 一 学期）

题 目： 计算器实现

|  |  |
| --- | --- |
| **专 业** | **计算机科学与技术** |
| **学 生 姓 名** | **梁佩欣** |
| **班 级 学 号** | **B18030409** |
| **指 导 教 师** | **易云山** |
| **指 导 单 位** | **计算机科学与技术系** |
| **日 期** | **2020.10.21** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **评 分 细 则** | **评分项和课程目标** | | | |
| 实验报告和程序完成情况能够反映出该生有兢兢业业的学习态度和责任感 | | **课程目标1（10分）** | |
|  | |
| 实验报告的撰写认真程度、规范性和完整性能够真实的反映出该生独立完成本次大作业，并且工作量饱满 | | **课程目标1（10分）** | |
|  | |
| 实验报告能够反映出该生掌握移动互联网的基本概念，对移动互联网最新应用开发技术有一定的了解 | | **课程目标1（10分）** | |
|  | |
| 从实验报告和程序完成情况反映出该生对课题中存在的复杂工程问题进行了分析并给出合理的解决方案 | | **课程目标1（30分）** | |
|  | |
| 从程序完成情况能够反映出该生熟练掌握最新移动开发应用技术的开发环境搭建、调试方法和接口的使用 | | **课程目标2（20分）** | |
|  | |
| 程序能够正确运行，满足功能要求，用户体验较好，并具备一定的创新性 | | **课程目标2（20分）** | |
|  | |
| **最终得分** |  | | |
| **评阅教师** | | **易云山** | **评阅日期** | **2020.XX.XX** |

**计算器实现**

**一、课题内容和要求**

**计算器的实现要求如下：**

1. 基本运算功能，至少支持十进制运算：加、减、乘、除、带（）运算表达式、平方、开方等；至少支持十六进制加、减、乘、除运算。
2. 科学计算器，至少实现：复数基本运算功能。
3. 支持房贷、车贷，最低要求：计算精度高、等额本金计算、等额本息计算等。
4. 当前汇率计算，至少实现：汇率的转换等。
5. 单位换算功能，至少实现：单位换算功能。
6. 自定义拓展功能
7. 功能需要在Android系统上实现演示。

**二、课题需求分析**

(1)本课题“计算器的功能框架”如下图1

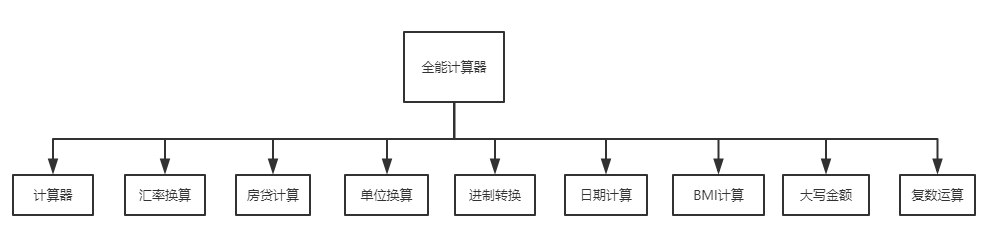


图1 功能框架图

(2)计算机模块实现的功能包括基础运算功能以及科学计数器。其中基础运算功能包含了加减乘除、平方、开方、带括号的运算，科学计算器中也包含了这些基础运算。

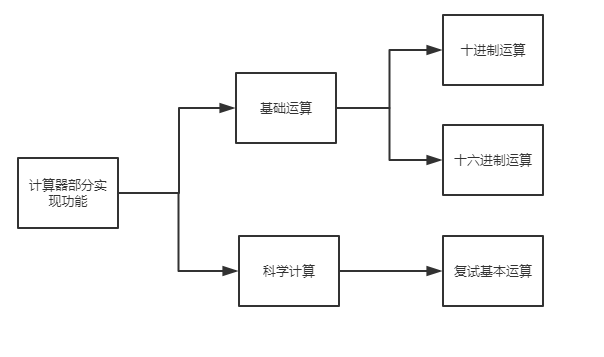


图2 计算器模块的功能

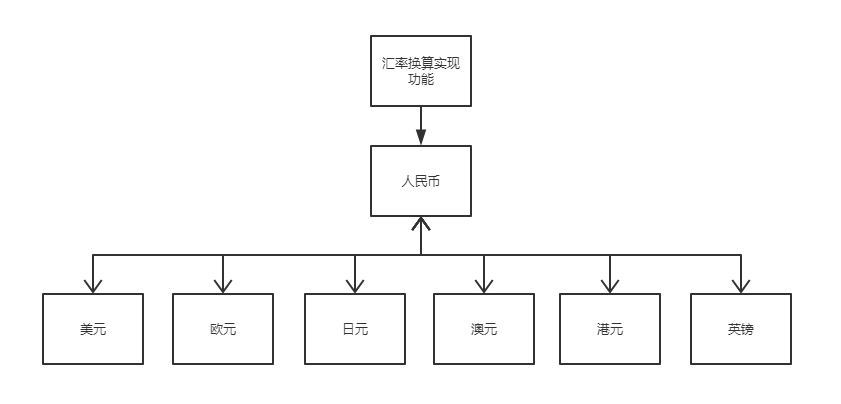
（3）汇率换算模块的功能实现了美元、欧元、日元、澳元、英镑和港元这六种与人民币之间的转换，同时还实现了同步更新基准汇率功能，在使用该计算器进行汇率换算时，会实时更新这六种币的基准汇率，以便求出最精确的汇率。

图3 汇率模块功能

（4）房贷计算，提供贷款年限，贷款利率以及贷款金额后可以根据等额本息和等额本金两种方式进行计算。在进行等额本息计算时会显示每期所还的本金和利息。进行等额本金计算时会也会同样显示每期所需的月供以及本金和利息相应值。

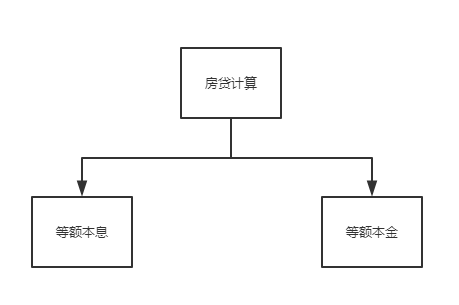


图4 房贷计算模块功能

单位换算功能实现了六大单位换算，分别是长度、时间、面积、重量、压强、速度。

●长度单位换算中可以实现毫米、厘米、分米、米、千米、英尺和英寸之间的转换；

●在时间单位换算中可以实现微秒、秒、分、时、天、周和年之间的转换；

●面积单位换算中分为公制、英制和市制三种模块，其中公制包括平方毫米、平方厘米、平方分米、平方米、公亩、公顷和平方千米，英制包括英亩、英里、平方码、平方英尺和平方英寸，市制中包括了顷、亩、分、尺和寸；

●重量单位换算中包含了毫克、克、千克、吨、磅、盎司、克拉、担、斤和两之间的换算；

●压强单位换算中包含了帕斯卡、百帕、千帕、兆帕、标准大气压、毫米汞柱、巴；

●速度单位换算中包含了米每秒、千米每秒、千米每小时、马赫、光速。

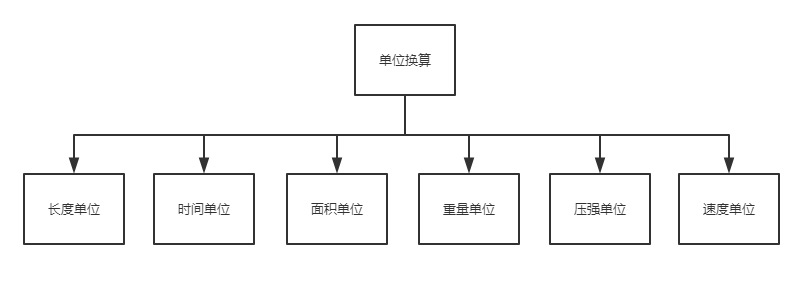


图5 单位换算模块功能

（6）进制转换实现了二进制、八进制、十进制和十六进制之间的转换。同时在此部分还实现了十六进制的基本运算功能

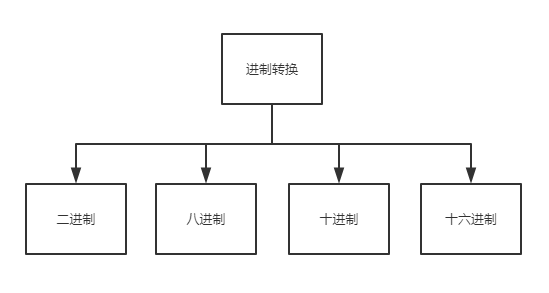


图6 进制转换模块功能

（7）BMI计算输入身高体重计算BMI，不在正常范围时会有“偏瘦”和“偏胖”的提醒。

（8）大写金额，输入一串数字相应的会换算出相应的大写。

（9）页面设计中实现了“我的工具”和“全部工具”的显示，可以将常用工具添加到“我的工具”里面，方便使用。同时还增加白天和夜间的功能，不同的时间段可以设置不同的背景显示，增加用户的体验。最后不仅做了Android 端的适配也做了ios的适配，手机系统不一样也不影响用户使用。

**三、程序设计**

1 主要数据结构

1. 房贷计算的算法：

function HousingLoan() {

const [amount, onChangeAmount] = useState('');

const [years, onChangeYears] = useState('');

const [rate, onChangeRate] = useState('4.9');

const [selectedMethod, onChangeMethod] = useState('key0');

const [results, setResults] = useState(undefined);

const AM = amount \* 10000; // AMount

const MR = (rate \* 0.01) / 12; // Monthly Rate

const months = years \* 12;

const compute = () => {

let r = results;

if (selectedMethod === 'key0') {

r = withEqualPrincipalAndInterest();

} else {

r = withEqualPrincipal();

}

setResults(r);

};

1. 大写金额算法：

function MoneyConversion() {

const [num, setNum] = useState('0');

const [upperNum, setUpperNum] = useState(digitUppercase(num));

useEffect(() => {

setUpperNum(digitUppercase(num));

}, [num]);

const compute = (item) => {

let temp = num;

switch (item) {

case '☒':

temp = temp.substr(0, temp.length - 1);

break;

default:

if (temp === '0' && item !== '.') {

temp = '';

}

temp = temp.concat(item);

}

setNum(temp);

};

const DATA = [

['7', '8', '9'],

['4', '5', '6'],

['1', '2', '3'],

['0', '.', '☒'],

];

1. 复数运算的算法：

function Plural() {

// 乘法

const [a1, onChangeA1] = useState('1');

const [b1, onChangeB1] = useState('1');

const [c1, onChangeC1] = useState('1');

const [d1, onChangeD1] = useState('1');

// 除法

const [a2, onChangeA2] = useState('1');

const [b2, onChangeB2] = useState('1');

const [c2, onChangeC2] = useState('1');

const [d2, onChangeD2] = useState('1');

// 结果

const [value1, setValue1] = useState('');

const [value2, setValue2] = useState('');

**四、源程序代码**

（1）房贷计算功能核心代码：

// 等额本金  
  const withEqualPrincipal = () => {  
    const totalInterest =  
      ((AM / months + AM \* MR + (AM / months) \* (1 + MR)) / 2) \* months - AM;  
    const totalRepayment = totalInterest + AM;  
    const details = [];  
    let principalPaid = 0; // 已归还本金累计额  
  
    for (let i = 0; i < months; i++) {  
      const repayment = AM / months + (AM - principalPaid) \* MR;  
      const interest = (AM - principalPaid) \* MR;  
      const principal = AM / months;  
      principalPaid += principal;  
      const item = {  
        index: `第${i + 1}期`,  
        repayment: repayment.toFixed(2),  
        principal: principal.toFixed(2),  
        interest: interest.toFixed(2),  
      };//每期所需的房贷  
      details.push(item);  
    }  
  
    return {  
      monthRepayment: details[0].repayment, // 首月月供  
      totalRepayment: totalRepayment.toFixed(2),  
      totalInterest: totalInterest.toFixed(2),  
      details,  
    };  
  };  
  
  // 等额本息  
  const withEqualPrincipalAndInterest = () => {  
    const monthRepayment =  
      (AM \* MR \* Math.pow(1 + MR, months)) / (Math.pow(1 + MR, months) - 1);  
    const totalRepayment = months \* monthRepayment;  
    const totalInterest = totalRepayment - amount;  
    const details = [];  
    for (let i = 0; i < months; i++) {  
      const interest =  
        (AM \* MR \* (Math.pow(1 + MR, months) - Math.pow(1 + MR, i))) /  
        (Math.pow(1 + MR, months) - 1);  
      const principal =  
        (AM \* MR \* Math.pow(1 + MR, i)) / (Math.pow(1 + MR, months) - 1);  
      const item = {  
        index: `第${i + 1}期`,  
        repayment: monthRepayment.toFixed(2),  
        principal: principal.toFixed(2),  
        interest: interest.toFixed(2),  
      };//每期所需房贷  
      details.push(item);  
    }

1. 大写金额的核心代码：

export const digitUppercase = function (n) {

var fraction = ['角', '分'];

var digit = ['零', '壹', '贰', '叁', '肆', '伍', '陆', '柒', '捌', '玖'];

var unit = [

['元', '万', '亿'],

['', '拾', '佰', '仟'],

];

var head = n < 0 ? '欠' : '';

n = Math.abs(n);

var s = '';

for (var i = 0; i < fraction.length; i++) {

s += (

digit[Math.floor(n \* 10 \* Math.pow(10, i)) % 10] + fraction[i]

).replace(/零./, '');

}

s = s || '整';

n = Math.floor(n);

for (var i = 0; i < unit[0].length && n > 0; i++) {

var p = '';

for (var j = 0; j < unit[1].length && n > 0; j++) {

p = digit[n % 10] + unit[1][j] + p;

n = Math.floor(n / 10);

}

s = p.replace(/(零.)\*零$/, '').replace(/^$/, '零') + unit[0][i] + s;

}

return (

head +

s

.replace(/(零.)\*零元/, '元')

.replace(/(零.)+/g, '零')

.replace(/^整$/, '零元整')

);

};

1. 复数运算的核心代码：

useEffect(() => {

const result = `${a1 \* c1 - b1 \* d1}+${b1 \* c1 + a1 \* d1}\*i`;

setValue1(result);

}, [a1, b1, c1, d1]);

useEffect(() => {

const result = `${

(a2 \* c2 + b2 \* d2) / (Math.pow(c2, 2) + Math.pow(d2, 2))

}+${(b2 \* c2 - a2 \* d2) / (Math.pow(c2, 2) + Math.pow(d2, 2))}\*i`;

setValue2(result);

}, [a2, b2, c2, d2]);

**五、测试数据及其结果分析**

对算法功能、性能以及健壮性评测所使用的输入数据进行介绍和说明，并给出与输入数据相匹配的算法执行结果，并进行分析。

（1）主页的展示：

图1 主页日间模式 图2主页夜间模式

（2）计算器功能测试：

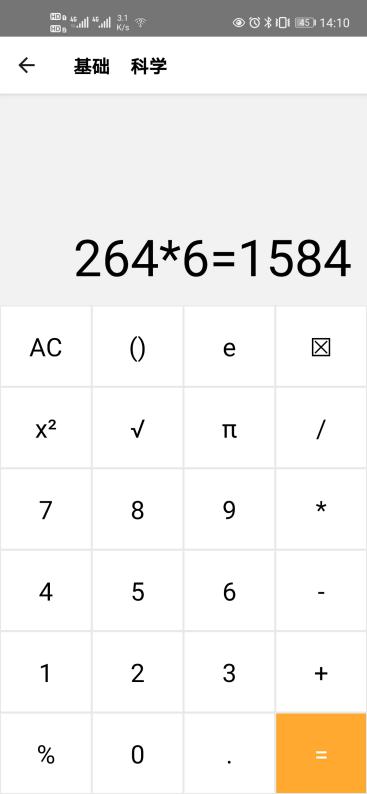
 

图3 计算器基本运算 图4 复数运算

（3）房贷和汇率测试

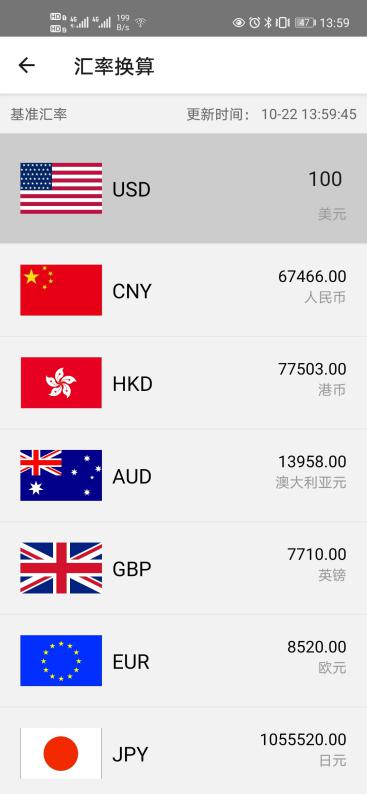
 

图5 房贷等额本息计算 图6 汇率功能测试

1. 单位换算功能

图7 长度单位测试 图8 重量单位测试

图9 面积单位测试 图10 时间单位测试

图11 压强单位测试 图12 速度单位测试

1. BMI功能和进制转换功能测试

图13 BMI功能测试 图14 进制转换测试

**六、课题完成过程中遇到的问题及解决方法**

问题1：

在进行房贷模块功能实现时，由于设置利率是以年利率设置的，只能调节年利率的大小，在房贷等额本金和等额本息问题中由于没有完全理解造成计算错误  
解决方法：

在房贷计算中需要自己先把月利率换成年利率，计算机才能正确计算出等额本息和等额本金的房贷问题。

问题2：

房贷计算在代码设计时遇到Android端和IOS系统适配不合适的问题，在Android端能正常显示，但在IOS端无法完全显示的问题。

解决方法：

在调节宽度时，由于IOS的系统板块设计问题，调节宽度不能像Android一样，一点一点的调出合适的宽度。

问题3：

房贷计算在进行日间和夜间模式时，调配出现问题。

解决方法：

房贷计算模块的字体之类的根据日间和夜间模式设置不同的颜色以及大小，方便查看。

**七、总结**

在为期一个多月的课程设计中更加深刻的理解了“嵌入式手机终端开发”这门课.我对这门课的设计有了更系统的认识。

个人感悟：在实践开始之前，我们进行了合理的分工，每个人都承担着这个计算器的一部分功能，在实验过程中，我们也是合作紧密，及时的沟通交流，互相解决不懂的问题。我们组的计算器不仅做了Android端的，还做了IOS系统的适配，由于这两个系统有些许差别，在代码设计过程中会遇到很多问题，比如格式和排版等。这些问题在我们的不断交流中得以解决，我们利用github这个网站进行交流，这样每个人的提交别人都会及时看到并更新。通过这次的课程设计，让我懂得了合作开发的重要性。合作不仅仅是一个人的事情，需要各方多交流才能找出问题所在。就比如在这一次的课程设计中，由于在设计的是两个系统，在一个系统完好展示时，很可能另一个系统不能展示。这就需要我们多交流。俗话说“三个臭皮匠赛过诸葛亮”，合作，积极沟通可以解决不少困惑自己的难题。

经验：

在课程设计中我们运用了JavaScript 语言进行应用开发，加上react可以实现交互。通过这次的课程设计，我对JS这门语言更加的熟悉和熟练。在这次的实验设计中，我对JS的知识不断深入理解，有一些知识点理解和实际应用是不一样的，这让我知道了我们需要多实践而不仅仅是理论学习。同时由于需要布局好看，在排版方面也有了很大的提高。在课程设计过程中，由于运用到React，学习了一门新技能，让自己更好的了解了前端。

建议：

1. 在进行功能模块设计时，可以参考网上别人的算法，或者是多思考，对自己的代码进行尽可能的优化，尽可能达到完美的效果。
2. 在遇到问题时先自己思考解决方案，不行的话可以询问队友，大家一起解决问题。
3. 学习一门新知识更多的是需要实际操作才能更好的了解它，计算机特别如此。所以在学习相应的知识点的时候最好自己去操作理解。