作业06 第6周案例EX401-EX405综合

# 一、作业要求

参照第6周案例EX401-EX405设计一个综合案例，案例包含6个页面，第1个页面为导航页，其他5个页面对应5个案例，利用第1个导航页可以进入这5个案例的页面。每个页面只是参考课程中的案例要求，案例内容不用和案例中的内容完全一致。

将案例代码和小程序的运行结果粘贴在下面相应的位置，运行结果以图片的形式粘贴（每个页面对应一张图片），最后将Word文档另存为PDF文件，并**将PDF文件单独上交，同时将小程序源代码压缩后单独上交。（上交的PDF文件类型正确，小程序源代码也全部提交，得10分）。**

# 作业内容

## 1．小程序代码（60分，粘贴小程序代码，包括各文件的文件名）

**只列出一些主要的代码**

### unit6work.wxml

<!--pages/unit6work/unit6work.wxml-->

<view class="box" style="color:white;background-color:#2b2b2b;margin:0;">

  <view class="title" style="color:white;">客官来点啥</view>

  <view class="outbox">

    <view class="box1">

      <navigator url="./chd/chd" style="width:50%;line-height:100rpx;">

        <view class="navbox">

          <view class="navtext">清蒸长度转换</view>

        </view>

      </navigator>

      <navigator url="../navmain/ovqe/ovqe" style="width:50%;line-height:100rpx;">

        <view>

          <view class="navtext">蒜炒一元二次方程</view>

        </view>

      </navigator>

    </view>

    <view class="box1">

      <navigator url="./cboxRadio/cBoxRadio" style="width:50%;line-height:100rpx;">

        <view>

          <view class="navtext">醋溜颜色条</view>

        </view>

      </navigator>

      <navigator url="../navmain/multmedia/multmedia" style="width:50%;line-height:100rpx;">

        <view>

          <view class="navtext">串烧音乐</view>

        </view>

      </navigator>

    </view>

    <view class="box1">

      <navigator url="./swiperTch/swiperTch">

        <view>

          <view class="navtext">凌晨3点，越肝越饿</view>

          <!-- <cover-image src="/images/right-arrow.png" style="width:40px;height40px"></cover-image> -->

        </view>

      </navigator>

    </view>

  </view>

  <view>

    <image src="/images/Squarepants.gif" mode="aspextFit"></image>

  </view>

</view>

### chd.wxml

<view class="box">

  <view class="title">长度转换</view>

  <form bindsubmit="calc" bindreset="reset">

    <input name="cd" type="digit" placeholder="输入长度（m）" value='{{chd}}'></input>

    <view class="btnlayout">

      <button type="primary" form-type="submit">计算</button>

      <button type="primary" form-type="reset">清除</button>

    </view>

    {{chdinput}}米等于

    <view class="waikuang">

      <view>微米：{{UM}}</view>

      <view>英尺:{{feet}}</view>

      <view>英寸:{{\_in}}</view>

      <view>码：{{yd}}</view>

      <view>英里：{{mile}}</view>

      <view>海里：{{nmi}}</view>

      <view>丈：{{zhang}}</view>

      <view>手：{{hand}}</view>

    </view>

  </form>

</view>

### chd.js

var m;

Page({

  /\*\*

   \* 页面的初始数据

   \*/

  data: {

  },

  calc: function(e) {

    // console.log("e.detail.value = ", e.detail.value.cd)

    m = parseFloat(e.detail.value.cd);

    console.log("m =", m)

    this.setData({

      chdinput: m,

      UM: m \* 1e6,

      feet: m \* 3.28084,

      \_in: m \* 39.37008,

      yd: m \* 1.093613,

      mile: m \* 0.000621371,

      nmi: m \* 0.000539957,

      zhang: m \* 0.333333,

      hand: m \* 5.36

    })

  },

  reset: function() {

    this.setData({

      chdinput: '',

      UM: '',

      feet: '',

      \_in: '',

      yd: '',

      mile: '',

      nmi: '',

      zhang: '',

      hand: ''

    })

  },

### ovqe.wxml

<view class="box">

  <view class="title">解一元二次方程</view>

  <form bindsubmit="formsubmit" bindreset="reset">

    <view class="layout">

      <input name="a" type="number" placeholder="a" placeholder-style="text-align:center" ></input>

      <text>x^2+</text>

      <input name="b" type="number" placeholder="b" placeholder-style="text-align:center" ></input>

      <text>x+</text>

      <input name="c" type="number" placeholder="c" placeholder-style="text-align:center" ></input>

      <text>=0</text>

    </view>

    <view class="layout">

      <button type='primary' form-type="submit">求根</button>

      <button type='primary' form-type="reset">清除</button>

    </view>

    <view class="box">{{result}}</view>

  </form>

</view>

### ovqe.js

var a, b, c, delta, x1, x2;

Page({

  /\*\*

   \* 页面的初始数据

   \*/

  data: {

    result: 'Hello, world!'

  },

  formsubmit: function(e) {

    a = parseFloat(e.detail.value.a);

    //a = parseFloat('');

    b = parseFloat(e.detail.value.b);

    c = parseFloat(e.detail.value.c);

    if (e.detail.value.b == '') {

      b = 0;

    }

    if (e.detail.value.c == '') {

      c = 0;

    }

    delta = b \* b - 4 \* a \* c;

    // this.setData({

    //   result: a

    // })

    // 无输入则abc为null（object类型），判断不能用==null

    // !a==0好像也不行，有0会被判true？

    if (e.detail.value.a == '') {

      this.setData({

        result: '熬夜多了连参数都忘了输了？'

      })

    } else if (a == 0) {

      wx.showToast({

        icon: 'fail',

        title: 'a=0！非二次方程',

        duration: 2000

      })

    } else if (delta < 0) {

      this.setData({

        result: 'delta<0，无实根！'

      })

    } else {

      let temp = Math.sqrt(delta);

      x1 = (-b + temp) / (2 \* a);

      x2 = (-b - temp) / (2 \* a);

      this.setData({

        result: 'x1为：' + x1 + '， x2为：' + x2

      })

    }

  },

  reset: function(e) {

    this.setData({

      result: '加油！'

    })

  },

### cBoxRadio.wxml

<!--pages/unit6work/cboxRadio/cBoxRadio.wxml-->

<view class="box">

  <view class="title">设置渐变</view>

  <checkbox-group bindchange="checkboxchange">

    <checkbox value="eblack">黑</checkbox>

    <checkbox value="eblue">蓝</checkbox>

    <checkbox value="ered">红</checkbox>

    <checkbox value="egreen">绿</checkbox>

    <checkbox value="ewhite">白 (两种以上才会显示)</checkbox>

  </checkbox-group>

  <radio-group bindchange="radiochange">

    <radio value="0deg">水平</radio>

    <radio value="60deg" checked="true">60°</radio>

    <radio value="45deg">45°</radio>

    <radio value="90deg">竖直</radio>

  </radio-group>

  <view class="gradientblock" style="background-image: linear-gradient({{degree}}, {{color}});"></view>

</view>

### cBoxRadio.js

  data: {

    degree:'60deg'

  },

  checkboxchange: function(e) {

    var colors = [];

    var ch = '';

    colors = e.detail.value;

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

    //   if('eblack' in )

    // }

    for(var i in colors){

      if(colors[i] == 'eblack'){

        ch +='black,'

      }

      if (colors[i] == 'eblue') {

        ch +='#0094ff,'

      }

      if (colors[i] == 'ered') {

        ch +='#ff0094,'

      }

      if (colors[i] == 'egreen') {

        ch +='#94ff00,'

      }

      if (colors[i] == 'ewhite') {

        ch +='white,'

      }

    }

    // ch -= ch[ch.length-1];  /\*\*减去末尾 逗号 \*/

    // ch -= ch[-1];  /\*\*减去末尾 逗号 \*/

    ch = ch.slice(0,-1);

    console.log("颜色ch",ch);

    this.setData({

      color:ch

    })

  },

  radiochange: function(e) {

    this.setData({

      degree:e.detail.value,

    })

    console.log("角度" ,e.detail.value);

  },

### multmedia.wxml

<view class="box">

  <view class="title">音乐播放器 </view>

  <view>

    <image src="/icons/music.png"></image>

  </view>

  <slider data-vol="vol" value="{{vol}}" max="1" step='0.01' bindchanging="volChanging" show-value="true" block-color="#1296db" activeColor="#1296db"></slider>

  <view>

    <button bindtap="pause\_continue">

      <image style="width:50px;height:50px;margin:auto auto;" mode="aspectFit" src="{{pc\_imageSrc}}"></image>

    </button>

  </view>

</view>

### multmedia.js

const audioContext = wx.createInnerAudioContext();

audioContext.volume = 0.5

audioContext.src = "/audio/PatrickStar.mp3"

// var flag =0

audioContext.onPlay(() => {

  console.log('开始播放')

}),

audioContext.onPause(()=>{

  console.log('播放暂停')

  // imageSrc =

})

audioContext.onError((res)=>{

  console.log(res.errMsg)

  console.log(res.errCode)

})

Page({

  /\*\*

   \* 页面的初始数据

   \*/

  data: {

    vol:audioContext.volume,

    songselect:0,

    // imageSrc:["/icons/play.png", "/icons/parse.png"],

    pc\_imageSrc: "/icons/play.png",

    // flag:flag

  },

  volChanging(e){

    // let vol = e.currenTarget.dataset.vol

    let value = e.detail.value;

    console.log('音量：',value)

    audioContext.volume = value

    // this.setData({

    //   vol:value

    // })

  },

  pause\_continue:function(e){

    console.log("进入函数 || 暂停状态为",audioContext.paused)

    if(audioContext.paused){

      audioContext.play()

      console.log('play')

      this.setData({

        // flag:1,

        pc\_imageSrc: "/icons/pause.png"

      })

      // console.log("setData之后 flag",flag)

    }

    if( !audioContext.paused ){

      audioContext.pause()

      console.log('pause')

      this.setData({

        // flag: 0,

        pc\_imageSrc: "/icons/play.png"

      })

      // console.log("setData之后 flag", flag)

    }

    console.log("函数结束 || 暂停状态为", audioContext.paused)

  },

  /\*\*

   \* 生命周期函数--监听页面加载

   \*/

  onLoad: function (options) {

    // audioContext.play()

    console.log('页面加载onLoad')

  },

  /\*\*

   \* 生命周期函数--监听页面初次渲染完成

   \*/

  onReady: function () {

    console.log('页面初次渲染onReady')

  },

  /\*\*

   \* 生命周期函数--监听页面显示

   \*/

  onShow: function () {

    console.log('页面显示onShow')

    console.log('暂停？',audioContext.paused)

  },

  /\*\*

   \* 生命周期函数--监听页面隐藏

   \*/

  onHide: function () {

    console.log('页面隐藏onHide')

    audioContext.stop()

  },

  /\*\*

   \* 生命周期函数--监听页面卸载

   \*/

  onUnload: function () {

    audioContext.stop()

    console.log('页面卸载onUnload')

  },

### swiperTch.wxml

<view class="box">

  <view class="title">Swiper and Switch</view>

  <swiper

  indicator-dots

  autoplay="{{auto}}"

  circular="{{circular}}"

  vertical="{{ver}}"

  interval="{{intervaltime}}"

  duration="500">

    <view wx:for="{{images}}" wx:key="{{index}}">

      <swiper-item>

        <image style="width:;height:;" mode="scaleToFill" src="{{item}}"></image>

      </swiper-item>

    </view>

  </swiper>

  <view class="waikuang">

    <text class="myleft">自动播放</text>

    <switch checked="{{auto}}" bindchange="changeAuto"></switch>

  </view>

  <view class="waikuang">

    <text class="myleft">循环播放</text>

    <switch checked="{{circular}}" bindchange="changeCircular"></switch>

  </view>

  <view class="waikuang">

    <text class="myleft">竖直播放</text>

    <switch checked="{{ver}}" bindchange="changeVer"></switch>

  </view>

  <view class="waikuang">

    <text class="myleft">自动切换时间间隔更快</text>

    <switch checked="{{interval}}" bindchange="changeInterval"></switch>

  </view>

</view>

### swiperTch.js

  data: {

    images:['/images/拼盘.png','/images/拼盘2.png','/images/白桃树莓冰.png'],

    auto:true,

    circular:false,

    ver:false,

    interval:false,

    intervaltime:5000

  },

  changeAuto:function(e){

    this.setData({

      auto: !this.data.auto,

      // auto: !this.auto

    })

  },

  changeCircular:function(e){

    this.setData({

      circular: !this.data.circular

    })

  },

  changeVer:function(e){

    this.setData({

      ver: !this.data.ver

    })

  },

  changeInterval:function(e){

    this.setData({

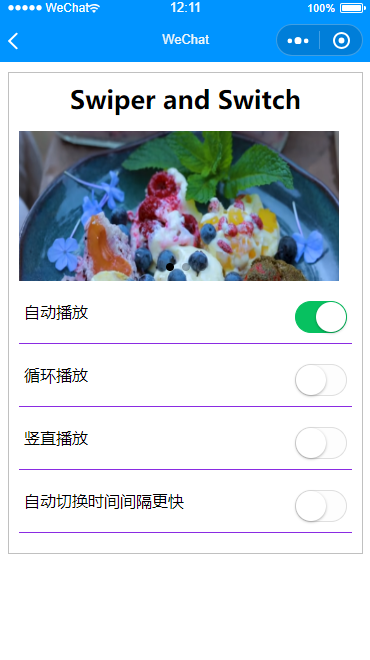
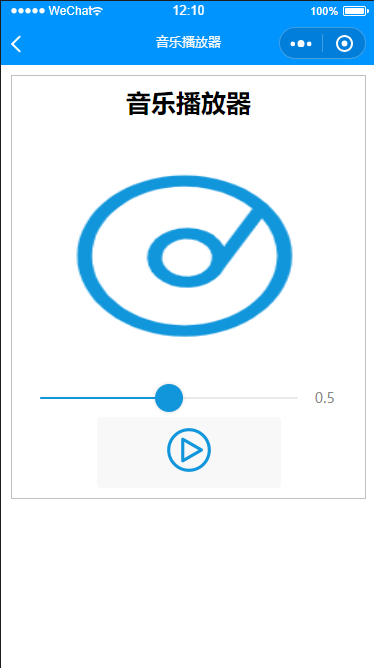
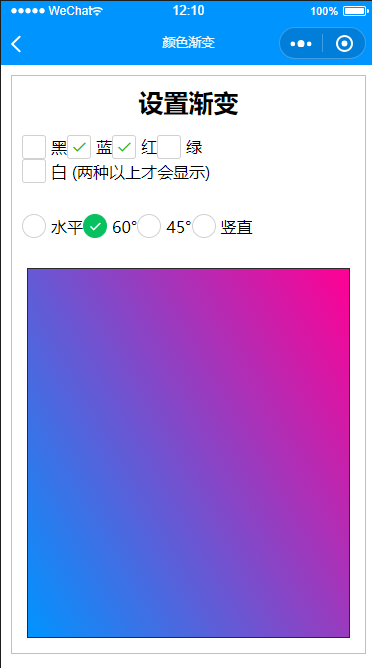
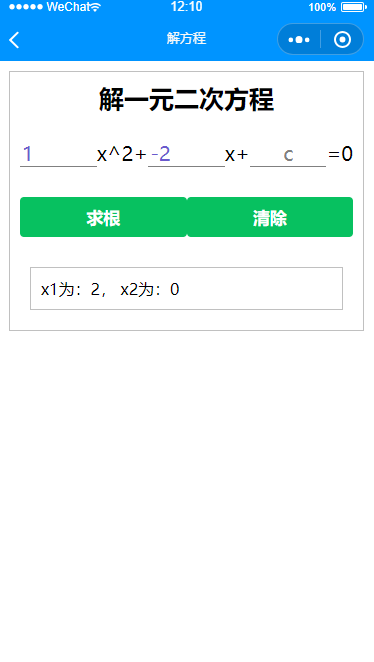
      interval:!this.data.interval,

      intervaltime: (this.intervaltime+2500)%5000

    })

  },

## 2. 案例的运行结果（20分，粘贴小程序运行结果图片，可以多屏）

****

## 小程序的主要设计思路（10分，把小程序的主要设计思路写在下面）

设计页面包括：

* **主菜单导航页、**

就是navigator导航

* **长度转换、**

参照案例中的货币，做的是把米转换成其他的长度单位，使用form表单提交、清除，提交后的参数先转成浮点型再按照转换比率计算

* **解一元二次方程、**

同样form表单提交数据，先判断参数条件是否为空，a必有，b、c可无，然后判断delta，再通过求根公式求解

* **设置颜色条、**

checkbox为多选，可共存；radio为单选，选项互斥

演示模块使用background-image:linea-gradient()设置渐变，使用checkbox-group设置添加颜色，使用radio-group设置颜色渐变角度。

* **音乐播放器、**

使用slider调节音乐声音大小（此外还可以调节播放进度）

* **轮播图与开关选择器**

和案例差不多，有一个开关改成了设置播放快慢