

Android第二次课--作业

- 请注意作业提交的邮件格式!!! 如果未按照需求提交的作业, 按未收到处理;
- 邮件提交的主题: 第x次-levelX-你的姓名-学号;
- 邮件内容: 你的github代码仓库地址;
- 提交到mredrock@163.com, 所有作业请使用java, idea完成;
- 请遵循代码的格式。
- **交作业时间:** 周天 (11月1日) 晚上24:00前

关于第二次课

如果类和对象以及面向对象的三大特性没弄懂的话, 下去多看看书, 也可以私聊我和其他管理, 加油, 坚持下去。



这是一个文字界面的游戏, 游戏信息使用命令框内输入输出

核心需求

整个游戏分为 (人物, 怪物的信息输入), 战斗, 战斗结束

游戏开始

需要提供游戏的主角名字和各项属性的输入

实例:

请输入主角名称:

...

请输入主角攻击力:

...

玩家需要输入的信息:

属性	字段类型及约束	说明
名称	String (1-50)	主角的名称
生命值	int (1-999)	主角的初始生命值
攻击力	int (1-999)	主角的攻击力（可自己配置装备）
防御力	int (1-999)	主角的防御力越大，收到的伤害越低（可自己配置装备）

怪物需要输入的信息：

属性		字段类型及约束说明
名称	String (1-50)	怪物的名称
生命值	int (1-999)	怪物的初始生命值
攻击力	int (1-999)	怪物的攻击力（可自己配置装备）

战斗阶段

战斗为回合制，回合数自己定义。

主角先攻击，然后怪物攻击，然后主角攻击，然后怪物攻击。（攻击是交替进行的，先主角，在怪物）

如果你有更好的攻击模式，可以自行设计，到时候在代码里面进行注释说明

每次回合结束输出战斗信息 例如：

小明同学对哥布林造成10点伤害
如果设置了防御值的话，伤害 = 攻击力 - 防御力。伤害最低为1

战斗结束

输出最后的主角和怪物的血量

这个作业其实蛮有意思的，可以更好的理解面向对象的概念，类，对象，封装，继承的使用等等。

你们也可以自行扩展，可以设计的很简单，也可以设计的很难。

Lv0:

复习上课所讲的知识，结合教程安装好git和注册github

Lv1:

- 设计主角类以及怪物类，两个类的属性（tips：体现封装）
- 能够输入主角和怪物类的属性

```
Scanner scanner = new Scanner();  
//输入整形  
Int atk = scanner.nextInt();  
//输入字符串  
String name = scanner.nextLine();
```

Lv2:

完成上面的战斗过程，战斗过程自行设计。

Lv3:

前面都是主角对怪物一对一战斗模式，根据上次课的作业，如果你们了解了ArrayList这个集合的概念，就可以添加多个怪物，进行一对多战斗模式。

关于一对多模式，主角 攻击 怪物的攻击模式以下三个自己进行选择（可以选择一个，也可以多个）：

- 随机random攻击一个怪物（给每个怪物编一个序号）
- 每个怪物都攻击一下
- 指定攻击怪物

知识点

- ArrayList
- 结合这节课知识
- 试试用图形界面表示最后的血条

Lv4:

在战斗过程中加入接口与回调的内容进行设计。（预习后在做）

例子

通过接口，当人接受到攻击后，生命力下降，如果没有死亡就通过回调，人进行反击，如果死亡了就结束战斗；怪物也是如此。

把攻击做成接口，把攻击力传过来（伤害=攻击力-防御）

知识点:

- 接口与回调
- 结合这节课知识

第二次课补充

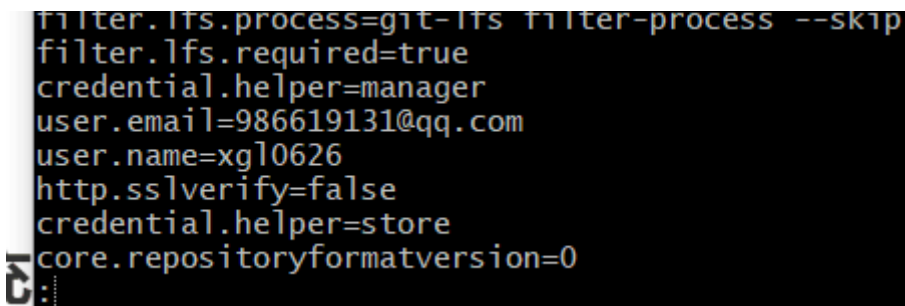
git

看到这里我相信你们都注册了github账号和密码了吧。

具体的流程我还是补充一下，就是首先在看到这个文档时，应该是结合安装教程（或者其他教程）已经安装好了git。

1.配置用户信息

```
git config --global user.name "账号名"//github用户名
git config --global user.email "邮箱" //github绑定的邮箱
git config --list 查看配置结果，可以看到github绑定的邮箱和github用户名
```



```
filter.lfs.process=git-lfs filter-process --skip
filter.lfs.required=true
credential.helper=manager
user.email=986619131@qq.com
user.name=xgl0626
http.sslverify=false
credential.helper=store
core.repositoryformatversion=0
```

2.上传代码

```
git init // 在你的项目文件里生成一个本地仓库

git remote add origin https://github.com/xgl0626/demo.git 填写你的远程仓库地址，这里是
我的远程仓库地址 //本地仓库关联远程仓库

git remote -v //查看当前关联的远程仓库

git add . //添加你写的代码文件

git commit -m "注释语句" （引号里写关于你代码的注释语句） //提交你写的代码文件

git push //推送你的代码文件到远程仓库，这一步基于你已经关联远程仓库才可以推送代码，否则会失
败

git push --set-upstream origin master //创建一个远程master分支并提交代码到远程仓库
master分支上
```

关于第一次push出现的问题;

1.如果远程仓库没有分支，就会出现下面的情况（分支的概念不懂没关系，先就这样用就行）

```
C:\Users\hp\IdeaProjects\test>git push
fatal: The current branch master has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin master
```

根据提示它的提示使用 `git push --set-upstream origin master` 即可。

2.提交的时候可能会输入你的github账号的密码，输入即可，或者了解一下教程里面的ssh密钥，关于密钥ssh与每次输入密码我还是比较推荐ssh。

static关键字（静态）

简单理解就是：方便在没有创建对象的情况下来进行调用（方法/变量）。

static关键字修饰的方法或者变量不需要依赖于对象来进行访问，只要类被加载了，就可以通过类名去进行访问。

所以说那个课上的那个方法结构其实是可以没必要加static的，下面这个更好理解，当时也是看到书上那么写，没有深究。

```
访问权限 返回值类型 方法名称（类型参数1,类型参数2, ...） {
//程序语句;
//[ return表达式];
}

public void eat (String food)
{

}

//这里只是简单提一下，更深的可以自行下去了解，也可以等下节课讲static
```