

# 实验六实验报告 吴禹 2023214309

---

## 作业题一

### 设计思路

根据题目的要求首先设计Hero类，类里面成员变量定义游戏主角需要的属性值。随后设计HeroMemento类用来存储某个时刻游戏主角的状态。最后设计CareTaker类将多个HeroMemento实例存储到本地，或者从本地读取存储的数据。

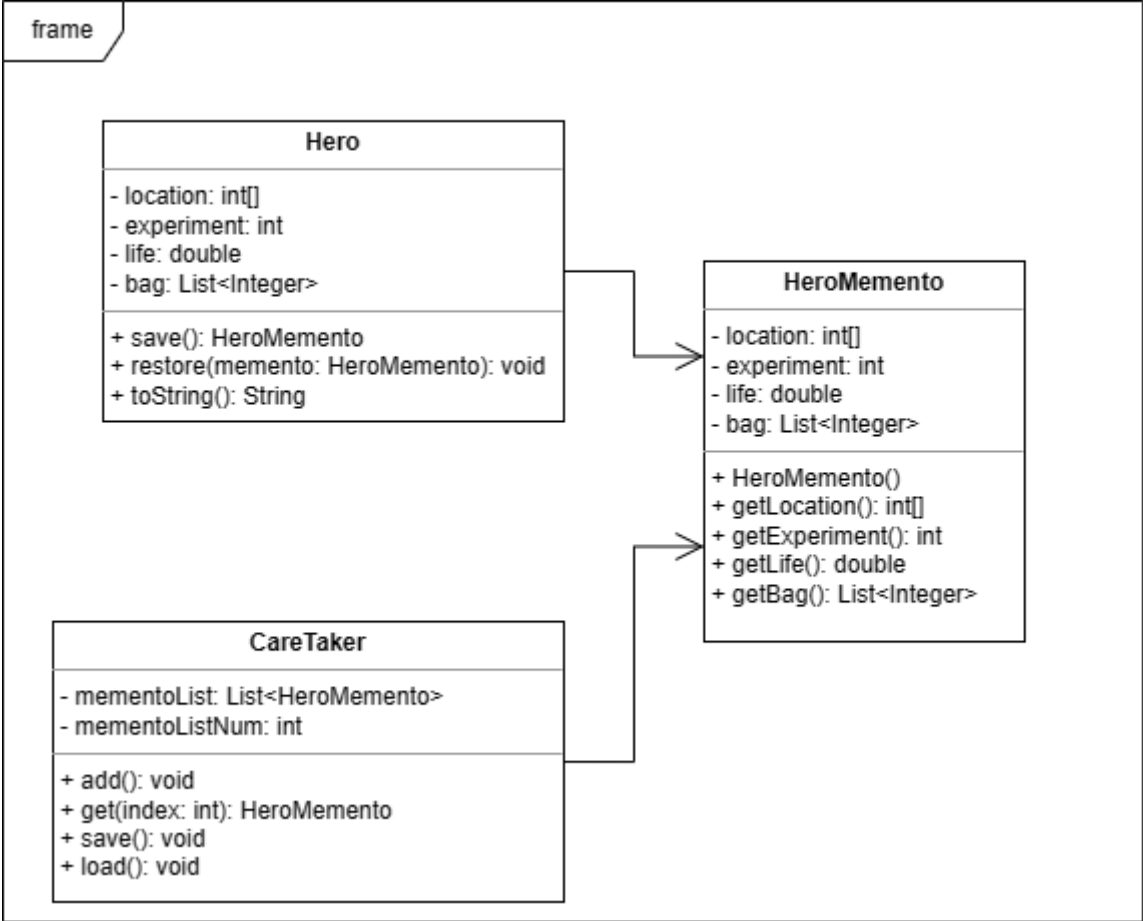
### 选用的设计模式

#### 备忘录模式

#### 关键类和函数的简单说明

- Hero 游戏主角类
  - location 位置信息
  - experiment 经验值
  - life 生命值
  - bag 背包
  - deleteItem() 删除背包的物品
  - addItem() 添加物品
  - save() 将当前的状态存储到HeroMemento类中
  - restore() 从HeroMemento实例中读取状态
  - toString() 将角色信息以字符串表示出来
- HeroMemento 存储角色某个时刻状态类
- CareTaker 存储所有记录的状态
  - mementoList 状态列表
  - mementoListNum 存储的状态数目
  - add() 添加状态
  - get() 根据下表获取状态
  - save() 将所有暂存的状态存储到本地文件里
  - load() 从本地文件读取存储的状态

#### UML类图



```
public static void main(String[] args){
    Hero hero = new Hero();
    CareTaker careTaker = new CareTaker();
    careTaker.load();
    if(careTaker.getMementoListNum() != 0){
        hero.restore(careTaker.get(careTaker.getMementoListNum() - 1));
    }
    hero.setLocation(new int[]{16, 110, 113});
    careTaker.add(hero.save());
    hero.setExperiment(999);
    careTaker.add(hero.save());
    hero.setLife(99.9);
    careTaker.add(hero.save());
    hero.addItem(789);
    hero.addItem(123);
    hero.addItem(234);
    hero.deleteItem(index: 1);
    hero.deleteItem(index: 1);
    hero.deleteItem(index: 1);
    careTaker.add(hero.save());
    careTaker.save();
}
```

## 输入输出

程序执行后的本地文件信息如下：

```
5 9 12 0 0.0
5 9 12 300 0.0
5 9 12 300 99.0
5 9 12 300 99.0 567,678
6 10 13 300 99.0 567,678
6 10 13 399 99.0 567,678
6 10 13 399 98.0 567,678
6 10 13 399 98.0 567,678,789,980
6 10 13 399 98.0 567,678,789,980
6 10 13 399 98.0 567,678,789,980
6 10 13 399 98.0 567,678,789,980
6 10 13 399 98.0 567,789,980,789,980
16 110 113 399 98.0 567,789,980,789,980
16 110 113 999 98.0 567,789,980,789,980
16 110 113 999 99.9 567,789,980,789,980
16 110 113 999 99.9 567,980,789,123,234
```

## 作业题二

### 设计思路

设计一个Play\_interface接口，里面定义视频播放器需要的组成元素。通过Video\_player类实现Play\_interface接口，实现组成元素具体的实现方式。最后根据三种模式具体需要的组成元素，设计三个类分别实现具体的要求。

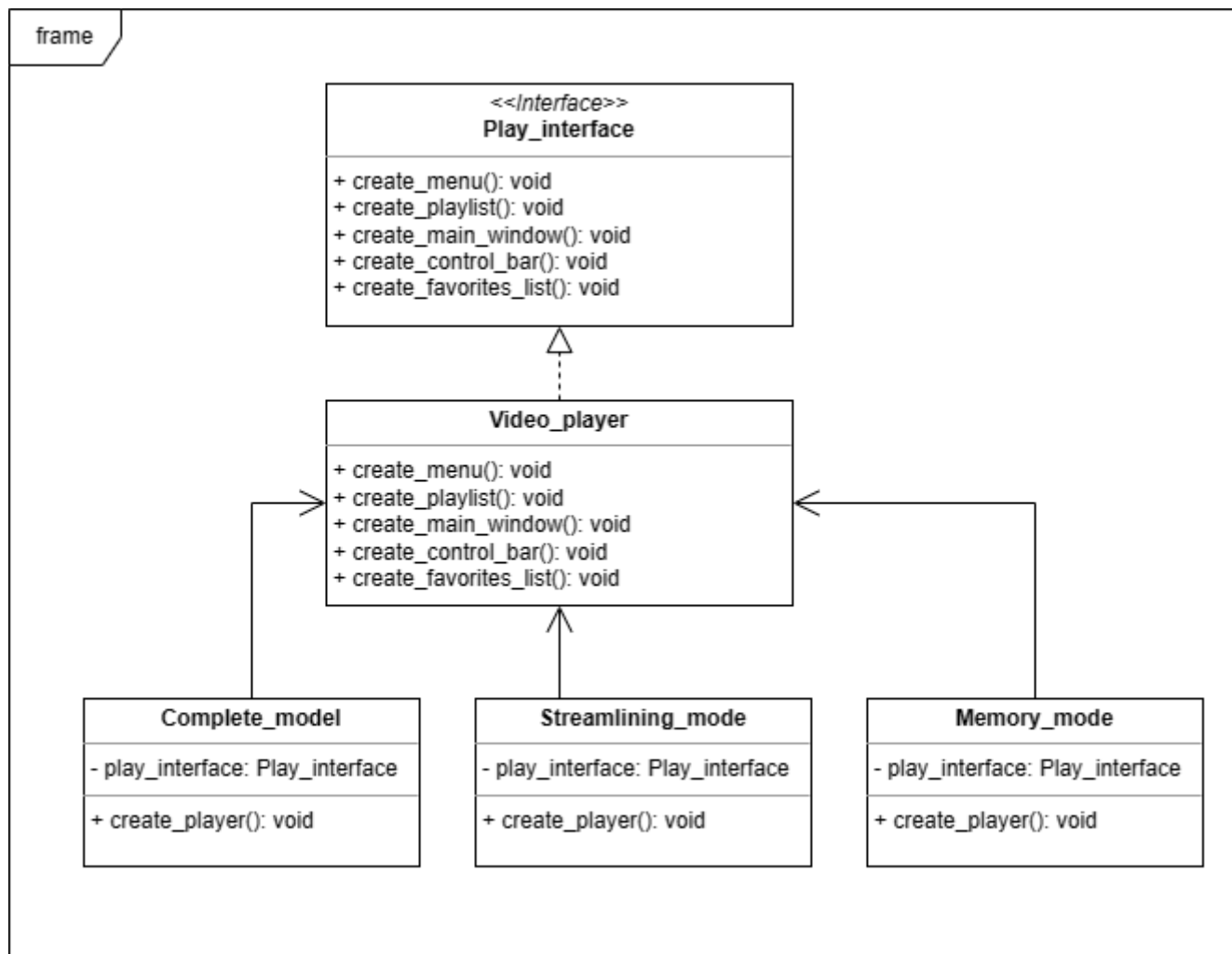
### 选用的设计模式

#### 建造者模式

#### 关键类和函数的简单说明

- Play\_interface 抽象组成元素的接口
  - create\_menu() 菜单
  - create\_playlist() 播放列表
  - create\_main\_window() 主窗口
  - create\_control\_bar() 控制条
  - create\_favorites\_list() 收藏列表
- Video\_player 实现接口定义的组成元素具体创建方法的类
- Complete\_model 实现具体模式的类
- Streamlining\_mode 实现精简模式的类
- Memory\_mode 实现记忆模式的类

### UML类图



## 测试用例

```

public static void main(String[] args){
    Video_player videoPlayer = new Video_player();
    Complete_model completeModel = new Complete_model(videoPlayer);
    completeModel.create_player();
    System.out.println("-----");
    Streamlining_mode streamliningMode = new Streamlining_mode(videoPlayer);
    streamliningMode.create_player();
    System.out.println("-----");
    Memory_mode memoryMode = new Memory_mode(videoPlayer);
    memoryMode.create_player();
}
  
```

## 输入输出

完整模式：  
显示菜单。  
显示播放列表。  
显示主窗口。  
显示控制条。

-----  
精简模式：  
显示菜单。  
显示控制条。

-----  
记忆模式：  
显示菜单。  
显示控制条。  
显示收藏列表。

进程已结束，退出代码为 0