设计模式实验

刘恒星 2022229044

April 3, 2023

1 访问者模式

实验要求

某软件公司需要设计一个源代码解析工具，该工具可以对源代码进行解析和处理，在该工具的初始版本中，主要提供了以下 3 个功能。

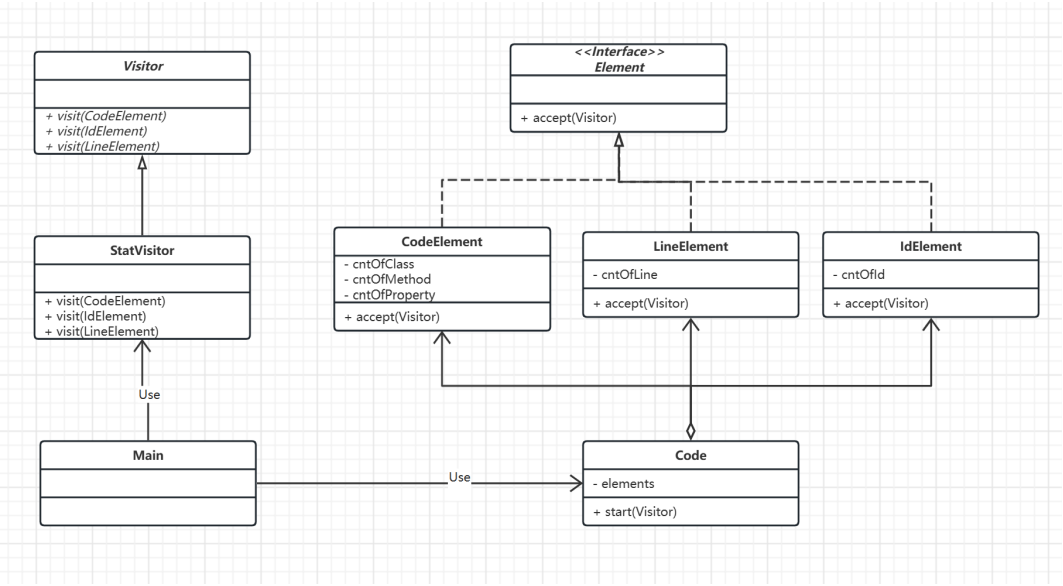
(1) 度量软件规模。可以统计源代码中类的个数、每个类属性的个数以及每个类方法的个数等。

(2) 提取标识符名称，以便检查命名是否合法和规范。可以提取类名、属性名和方法名等。

(3) 统计代码行数。可以统计源代码中每个类和每个方法中源代码的行数。

将来还会在工具中增加一些新功能，为源代码中的类、属性和方法等提供更多的解析操作。现采用访问者模式设计该源代码解析工具，可将源代码中的类、属性和方法等设计为待访问的元素，上述不同功能由不同的具体访问者类实现，绘制对应的类图并编程模拟实现。

类图设计



代码运行效果

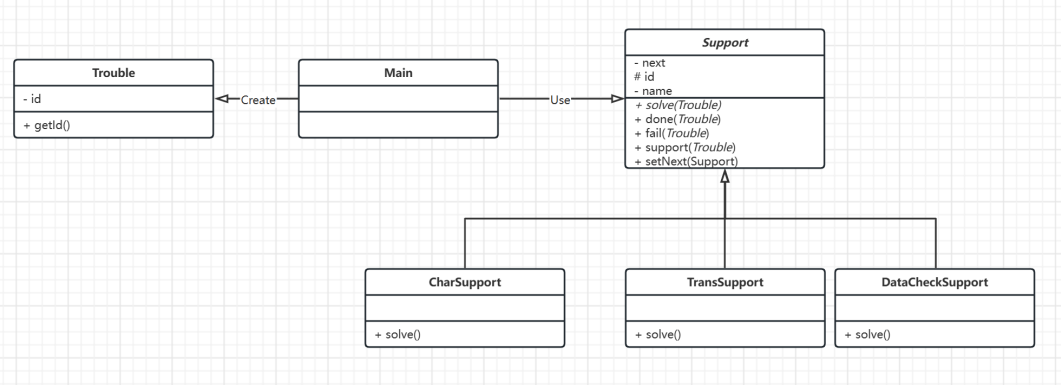
```
PS D:\Desktop\course\hw\design pattern\designPatternHW3\Visitor>  & 'C:\Program Files\Java\j
tern\designPatternHW3\Visitor\bin' 'App'
STAT: CodeElement [cntOfClass=4, cntOfMethod=20, cntOfProperty=3]
STAT: LineElement [cntOfLine=200]
STAT: IdElement [cntOfId=70]
PS D:\Desktop\course\hw\design pattern\designPatternHW3\Visitor>
```

2 职责链模式

实验要求

在某 Web 框架中采用职责链模式来组织数据过滤器，不同的数据过滤器提供了不同的功能，例如字符编码转换过滤器、数据类型转换过滤器、数据校验过滤器等，可以将多个过滤器连接成一一个过滤器链，进而对数据进行多次处理。根据以上描述，绘制对应的类图并编程模拟实现。

类图设计



代码运行效果

```
PS D:\Desktop\course\hw\design pattern\designPatternHW3\Chain>  d:; cd 'd:\Desktop
361\bin\java.exe' '-cp' 'D:\Desktop\course\hw\design pattern\designPatternHW3\Chai
Trouble [id=-1] is solved by Support [name=数据类型转换过滤器]
Trouble [id=0] cannot solve
Trouble [id=1] is solved by Support [name=字符编码转换过滤器]
Trouble [id=2] is solved by Support [name=数据类型转换过滤器]
Trouble [id=3] is solved by Support [name=数据校验过滤器]
```

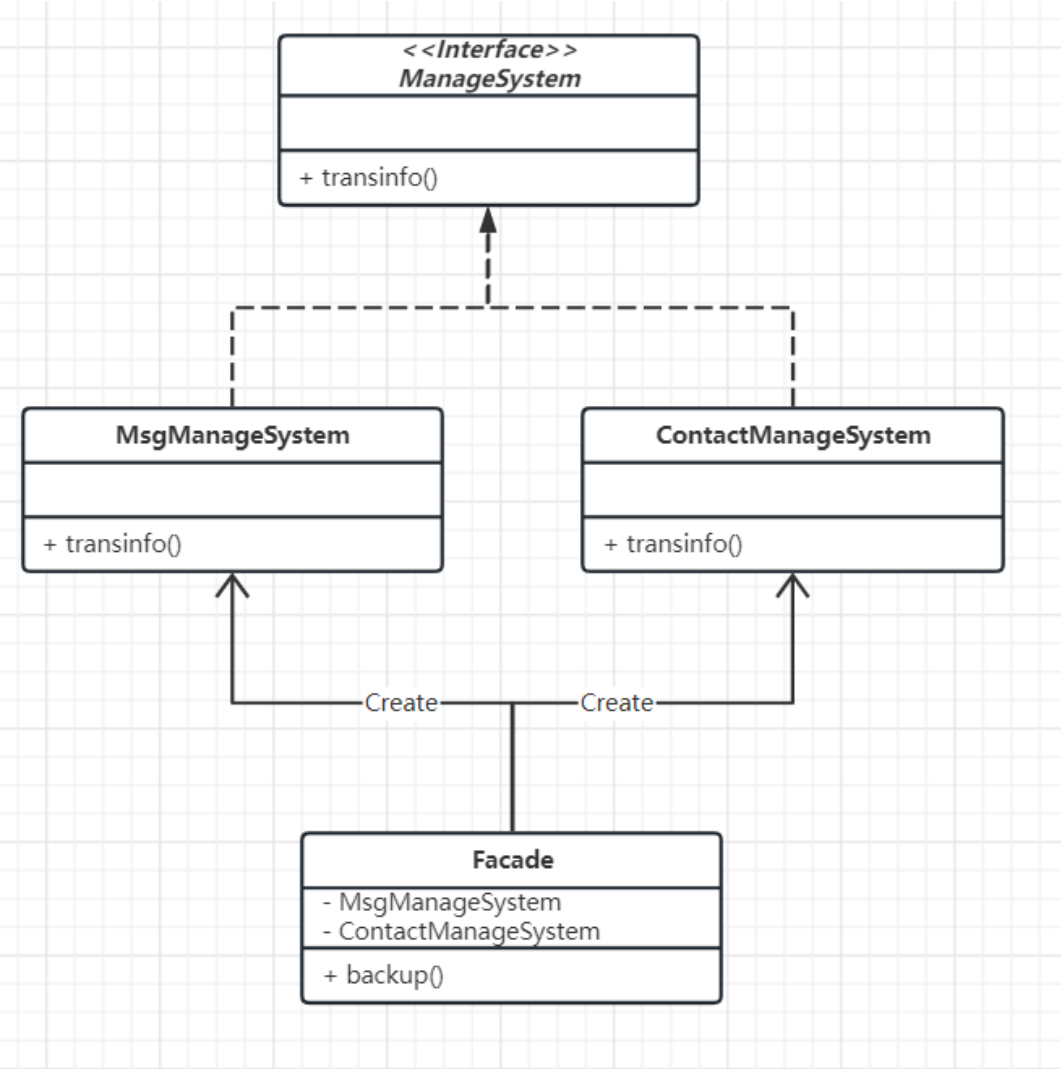
3 外观模式

实验要求

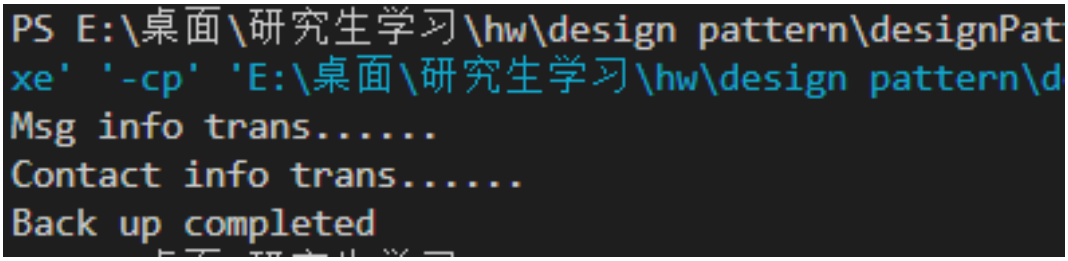
某软件公司为新开发的智能手机控制与管理软件提供了一键备份功能，通过该功能可以将原本存储在手机中的通讯录、短信、照片、歌曲等资料一次性全部拷贝到移动存储个质（例如 MMC

卡或 SD 卡）中。在实现过程中需要与多个已有的类进行交互，例如通讯录管理类、短信管理类等。为了降低系统的耦合度，试使用外观模式来设计并编程模拟实现该一键备份功能。

类图设计



代码运行效果

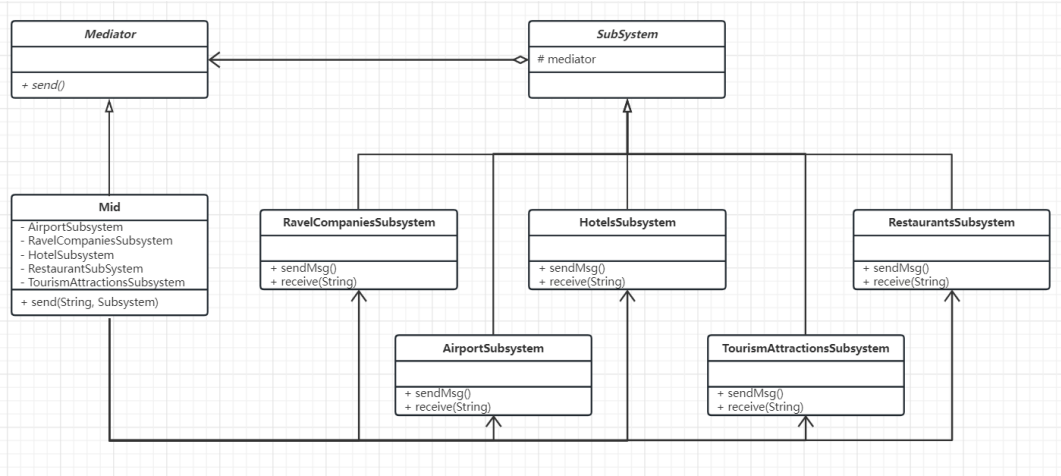


4 中介者模式

实验要求

为了大力发展旅游业，某城市构建了一个旅游综合信息系统，该系统包括旅行社子系统（ravel companies Subsystem）、宾馆子系统（HotelsSubsystem）、餐厅于系统（Restaurants Subsystem）、机场子系统（Airport Subsystem）、旅游景点子系统（Tourism Attractions Subsystem）等多个子系统，通过该旅游综合信息系统，各类企业之间可实现信息共享，一家企业可以将客户信息传递给其他合作伙伴。例如，当一家旅行社有一些客户后，该旅行社可以将客户信息传送到宾馆子系统、餐厅子系统、机场子系统和旅游景点子系统；宾馆也可以将顾客信息传送到旅行社子系统、餐子系统、机场子系统和旅游景点子系统；机场也可以将乘客信息传送到旅行社子系统、宾馆子系统、餐厅子系统和旅游景点子系统。由于这些子系统之间存在较为复杂的交互关系，现采用中介者模式为该旅游综合信息系统提供一个高层设计，绘制对应的类图并编程模拟实现。

类图设计



代码运行效果

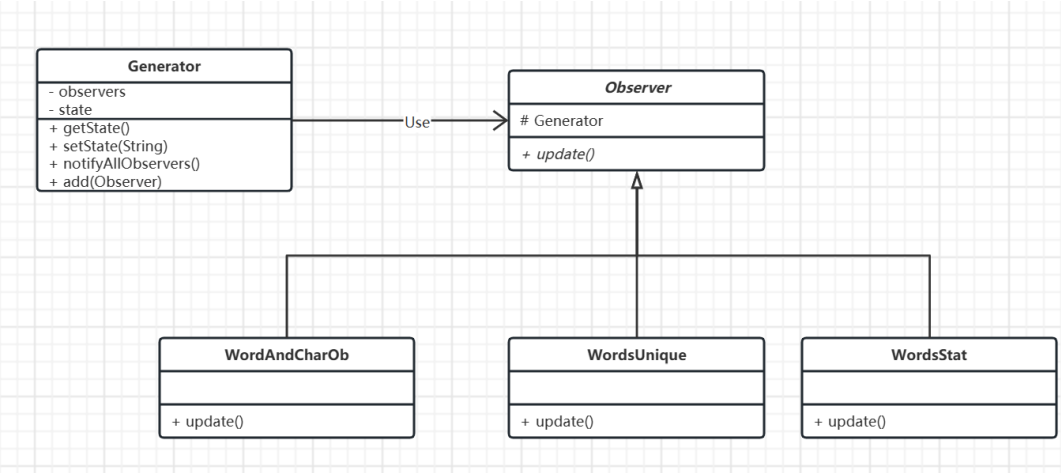
```
PS E:\桌面\研究生学习\hw\design pattern\designPatternHW3\Mediator>  & 'E:\java\bi
ator\bin' 'App'
Hotels receive: Msg of RavelCompaniesSubsystem
Restaurants  receive: Msg of RavelCompaniesSubsystem
Airport  receive: Msg of RavelCompaniesSubsystem
Tourism     Attractions  receive: Msg of RavelCompaniesSubsystem
==================
Ravel Companies receive: Msg of HotelsSubsystem
Restaurants  receive: Msg of HotelsSubsystem
Airport  receive: Msg of HotelsSubsystem
Tourism     Attractions  receive: Msg of HotelsSubsystem
==================
Ravel Companies receive: Msg of AirportSubsystem
Restaurants  receive: Msg of AirportSubsystem
Hotels receive: Msg of AirportSubsystem
Tourism     Attractions  receive: Msg of AirportSubsystem
==================
```

5 观察者模式

实验要求

某文字编辑软件须提供如下功能：在文本编辑窗口中包含一个可编辑文本区和 3 个文本信息统计区，用户可以在可编辑文本区对文本进行编辑操作，第一个文本信息统计区用于显示可编辑文本区中出现的单词总数量和字符总数量，第二个文本信息统计区用于显示可编辑文本区中出现的单词（去重后按照字典序排序），第三个文本信息统计区用于按照出现频次降序显示可编辑文本区中出现的单词以及每个单词出现的次数（例如 hello :5）。现采用观察者模式设计该功能，绘制对应的类图并编程模拟实现。

类图设计



代码运行效果

```
PS D:\Desktop\course\hw\design pattern\designPatternHW3\Observe>  & 'C:\Program Files\Java\jdk1.8.0_361\bin\java.exe'
tern\designPatternHW3\Observe\bin' 'App'
First update:Hello World!
WordAndCharOb: Words: 2 Char: 11
WordsUnique: [Hello, World!]
WordsStat: {Hello=1, World!=1}
Second update:This is TJU DesignPattern homework3. Author is 2022229044 LHX
WordAndCharOb: Words: 9 Char: 53
WordsUnique: [2022229044, Author, DesignPattern, Hello, LHX, TJU, This, World!, homework3., is]
WordsStat: {2022229044=1, Author=1, DesignPattern=1, Hello=1, LHX=1, TJU=1, This=1, World!=1, homework3.=1, is=2}
PS D:\Desktop\course\hw\design pattern\designPatternHW3\Observe>
```

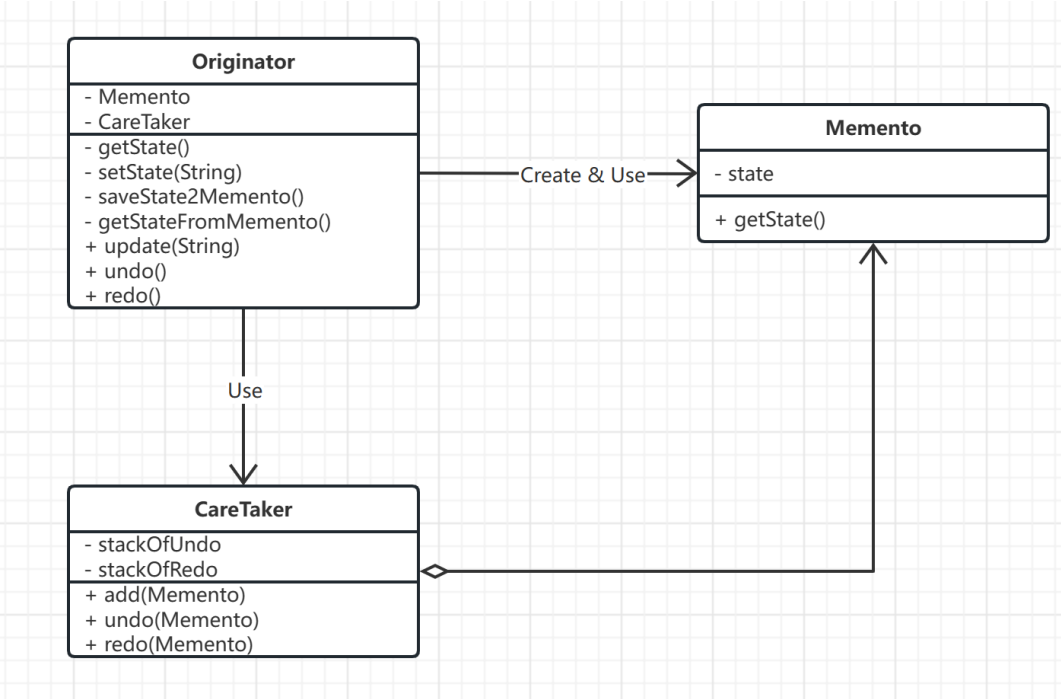
6 备忘录模式

实验要求

某文字编辑软件须提供撤销（Undo）和重做 / 恢复（Redo）功能，并且该软件可支持文档对象的多步撤销和重做。开发人员决定采用备忘录模式来实现该功能，在实现过程中引人栈（Stack）作为数据结构。在实现时，可以将备忘录对象保存在两个栈中，一个栈包含用于实现撤销操作的状态对象，另一个栈包含用于实现重做操作的状态对象。在实现撤销操作时，会弹出撤销栈栈顶

对象以获取前一个状态并将其设置给应用程序；同样，在实现重做操作时，会弹出重做栈栈顶对象以获取下一个状态并将其设置给应用程序。绘制对应的类图并编程模拟实现。

类图设计



代码运行效果

```
PS D:\Desktop\course\hw\design pattern\designPatternHW3\Memento>  d:; cd 'd:\Desktop\course\hw\design pattern\designPat
\Java\jdk1.8.0_361\bin\java.exe' '-cp' 'D:\Desktop\course\hw\design pattern\designPatternHW3\Memento\bin' 'App'
Originator [state=Hello World!]
After update: Originator [state=Hello World! I add some text]
After undo: Originator [state=Hello World!]
After redo: Originator [state=Hello World! I add some text]
```

7　感想

在这次实验中，学会了这 6 个设计模式的相关知识。初步掌握了这些设计模式的原理，并且能够编写代码。新的设计模式涉及到了新的行为型设计模式，设计的很奇妙，解决了我以前的一些困惑。