Tin Thu Zar Aye

Duc Ta

CSC 340 section 03 (Tic)

02/18/2020

Cohesion

Cohesion is a measure of how the method of a class is meaningfully and strongly related and how focused they are in providing a well-defined purpose to the system. A class should describe a single entity, and all the class operation should logically fit together to support a coherent purpose. A single entity with many responsibilities can be broken into several classes to separate the responsibilities. There are 2 different cohesions in java which are high cohesion and low cohesion.

Low Cohesions- A class that contains numerous that have nothing in common is an example of cohesion. Moreover, when a class contains many unrelated functions, it can be identified as a low cohesion class. For example,

```
class Check{
public void display(stirng s){}
public String getMethod(){}
}
```

High Cohesion- A class contain methods that serve a common purpose. The code has to be very specific in operations and the responsibilities are highly related to the class. The class should be small and with high purpose and related functions. High Cohesion class is easy to maintain and can evolve and grow in a controlled manner. High Cohesion class is measure of how the methods of a class is meaningfully and strongly related and how focused they are in providing a well-defined purpose to the system.

Instance Vs Static

Instance and Static are integral parts of object-oriented programming. A data filed or method is either instance or static.

Instance – Instance variable hold value that are associated with an individual object. A variable or method that is dependent on a specific instances of the class must be an instance variable or method. A constructor is always instance because it is used to create a specific instance. Moreover, instance variable is property of each object of a class. It can't be reinitialized directly within class. For example,

```
class Check{
int a = 10; // instance variable
double b = 10.99; // instance variable
}
```

Static – A static variable or method can be invoked form an instance method, but an instance variable or method can't be invoked from a static method. A variable that is shared by all the instances of a class should be declared static. Moreover, always reference static variable and methods from a class name (rather than a reference variable) to improve readability and avoid errors. It doesn't pass the parameter from a constructor to initialize a static data field. It is better to use the setter method to change the static data field. You can not mistakenly overlook static data fields or methods. It is a common design error to define and instance method should have been static. Static variable is property of the class itself, and doesn't belong to any particular object of that class.

Interface Vs Abstract Classes

Both interface and abstract classes can be used to specify common behavior for objects. In general, a strong is a relationship that clearly describes a parent-child relationship should be modeled using classes. The vitues of interfaces and abstract classes can be combines by creating an interface with an abstract class that implement it. Then you can use the interface ot the abstract class whichever is convenient. Both interface class and Abstract class are used for abstraction.

Interface- A weak is a relationship, also known as an is kind of relationship indicates that an object possesses a certain property. A weak is a relationship can modeled using interfaces. Interface are more flexible than abstract classes because subclass can extend only one superclass but can implement any number of interfaces. However, interfaces can't contain concrete methods. Moreover, variable declared in a interface are by default final and member of a java interface are public by default. A java interface can be implemented using keyword "implements". Interface can't provide the implementation of abstract class.

Abstract – An abstract class may contain non-final variables. Abstract class can have final, non-final, static and non-static variable. Abstract class can provide the implementation of interface. Abstract class can be extended using keyword "extends". Moreover, abstract class can extend another java class and implement multiple java interfaces. A java abstract class can have class members like private and protected.

Consistency

It follows standard Java programming style and naming conventions. It chooses informative names for classes, data field and methods. A popular style is to place the data declaration before the constructor and place constructors before methods. It makes the names consistent. It is not a good practice to choose different names for similar operations. In general, you should consistently provide a public no-arg constructor for constructing a default instance. If a class does not support no-arg constructor, document the reason. If no constructors are defined explicitly, a public default no-arg constructor with an empty body is assumed. If you want to prevent users from creating an object for a class, you can declare a private constructor in the class, as is the case for the Math class.

For example,

Clarity

Cohesion, consistency and encapsulation are good guidelines for achieving design clarity. Additionally, a class should have a clear contract that is easy to explain and easy to understand. User can incorporate classes in many different combinations, orders and environments. Therefore, you should design a class that imposes no restrictions on how or when the user can use it, design the properties in a way that lets the user set them in any order and with any combination of values and design methods that functions independently of their order of occurrence. Method should be defined intuitively without causing confusion. You should not declare a data filed that can be derived from other data fields.

PART B

1.I am solving the problem of the “Dictionary 340 JAVA”. We need to store 3 Strings which are keyword, speech and definition in the enum. The problem I am solving is when the user enters only one String for the search word, the program should be output the whole group of which related with the search key. In this problem, “distinct” use as a filter because when the user enters the 2 String, for example; “distinct distinct,” the second distinct use as a filter which means the output is only one line each of the definition. Moreover, when the user enters the 3 String, for example; “distinct noun distinct”, the output is only the noun (Speech) which has different definition because by including “distinct”, it is going to filter noun’s definition. There is also no case sensitive when the user enters the search key. It doesn’t matter that the user enter either lowercase of the search key or uppercase of the search key. The program end when the user say “!Q”

I store each data in enum class. And then I created the constructor, getter and setter for the enum class to access each String. In the main, I create the new object as temporary to access the data that I define in the enum class.

The most I use in this problem is “if statement” and “while loop”. It is true that there is many other ways to design the program but in my opinion, designing with “if statement” is easy to understand and easy to read the program for the other people.

2. Yes, my program is working properly.

Sample output

```
/*
! Loading data...
! Loading completed...

-----DICTIONARY 340 JAVA-----


Search:
arrow
[Ljava.lang.String;@41629346
Arrow [noun] : Here is one arrow: <IMG> -=>> </IMG
Search:
distinct
[Ljava.lang.String;@15aeb7ab
Distinct [adjective] : Familiar. Worked in Java
Distinct [adjective] : Unique. No duplicates. Clearly different or of a different kind.
Distinct [adverb] : Uniquely. Written "distinctly".
Distinct [noun] : A keyword in this assignment
Distinct [noun] : A keyword in this assignment
Distinct [noun] : A keyword in this assignment
Distinct [noun] : An advanced search option
Distinct [noun] : Distinct is a parameter in this assignment
Search:
```

```
distinct distinct

[Ljava.lang.String;@7b23ec81

Distinct [adjective] : Familiar. Worked in Java

Distinct [adjective] : Unique. No duplicates. Clearly different or of a different kind.

Distinct [adverb] : Uniquely. Written "distinctly".

Distinct [noun] : A keyword in this assignment

Distinct [noun] : An advanced search option

Distinct [noun] : Distinct is a parameter in this assignment

Search:

distinct noun distinct

[Ljava.lang.String;@6acbcfc0

Distinct [noun] : A keyword in this assignment

Distinct [noun] : An advanced search option

Distinct [noun] : Distinct is a parameter in this assignment

Search:

placeholder

[Ljava.lang.String;@5f184fc6

Placeholder [adjective] : To be updated...

Placeholder [adjective] : To be updated...

Placeholder [adverb] : To be updated...

Placeholder [conjunction] : To be updated...

Placeholder [interjection] : To be updated...

Placeholder [noun] : To be updated...
Placeholder [noun] : To be updated...

Placeholder [noun] : To be updated...

Placeholder [preposition] : To be updated...

Placeholder [pronoun] : To be updated...

Placeholder [verb] : To be updated...

Search:

placeholder adjective

[Ljava.lang.String;@3feba861

Placeholder [adjective ] : To be updated...

Placeholder [adjective ] : To be updated...

Search:

placeholder adjective distinct

[Ljava.lang.String;@5b480cf9

Placeholder [adjective] : To be updated...

Search:

placeholder distinct

[Ljava.lang.String;@6f496d9f

Placeholder [adjective] : To be updated...

Placeholder [adverb] : To be updated...

Placeholder [conjunction] : To be updated...

Placeholder [interjection] : To be updated...

Placeholder [noun] : To be updated...

Placeholder [preposition] : To be updated...
Placeholder [pronoun] : To be updated...

Placeholder [verb] : To be updated...

Search:

csc340 distinct

[Ljava.lang.String;@723279cf

CSC340 [adjective] :   = C++ version of CSC210 + CSC220 + more

CSC340 [noun] :  A CS upper division course.

CSC340 [noun] : Many hours outside of class

CSC340 [noun] : Programming Methodology.

Search:

csc340 noun

[Ljava.lang.String;@10f87f48

CSC340 [noun] :  A CS upper division course.

CSC340 [noun] : Many hours outside of class

CSC340 [noun] : Programming Methodology.

Search:

csc220 adjective distinct

[Ljava.lang.String;@b4c966a

CSC220 [adjective] : Ready to create complex data structures.

Search:

csc220 verb

[Ljava.lang.String;@2f4d3709

CSC220 [verb] : To create data structures.
```

```
Search:
BOOK distinct
[Ljava.lang.String;@4e50df2e
Book [noun] : A set of page.
Book [noun] : A written work published in printed or electronic form.
Book [verb] : To arrange for someone to have a seat on a plane.
Book [verb] : To arrange something on a particular date.
Search:
adverb noun distinct
[Ljava.lang.String;@1d81eb93
Adverb [noun] : Adverb is a word that adds more information about place, time, manner, cause or degree to a verb, an adjective, a phrase or another adverb.
Search:
adjective distinct
[Ljava.lang.String;@7291c18f
Adjective [noun] : Adjective is a word that describes aperson or thing, for example big, red and clever in a big house, red wine and a clever idea.
Search:
CONJUNCTION verb
[Ljava.lang.String;@34a245ab
<Not found>
Search:
interjection noun distinct
```

```
[Ljava.lang.String;@7cc355be
Interjection [noun] : Interjection is a shout sound, word or phrase spoken suddenly to express an emotion. Oh!, Look out! and Ow! are interjections.
Search:
noun noun
[Ljava.lang.String;@6e8cf4c6
Noun [noun] : Noun is a word that refers to a person, (such as Ann or doctor), a place (such as Paris or city) or a thing, a quality or an activity (such as plant, sorrow or tennis).
Search:
preposition prep distinct
[Ljava.lang.String;@12edcd21
<2nd argument must be part of speech or "distinct"
Search:
preposition preposition
[Ljava.lang.String;@34c45dca
Search:
verb verb distinct
[Ljava.lang.String;@52cc8049
<Not found>
Search:
facebook
[Ljava.lang.String;@5b6f7412
<Not found>
```

```
Search:

!Q

[Ljava.lang.String;@27973e9b

Thank you!!

*/
```