

Interfaces



Gruppe: Andreas Lind, Wajid, Ali.
Klasse: Dat18c.
Dato: 25-03-2019.

Interfaces

Interfaces work kind of like a storage point between methods from one class or several and another class in the sense that the interface stores methods from other classes to be used in other classes.

Interfaces also seems to be a good way to structure how a class is going to look because if you make a interface you kind of have to structure your class after that interface. Or you could do it the other way around to be sure that your interface is going to be right.

Interfaces in a physical sense seems to be use for sending input from one place to another but there are exceptions. [https://en.wikipedia.org/wiki/Interface_\(computing\)](https://en.wikipedia.org/wiki/Interface_(computing)) Microphones, and buttons on a tv.

Note

It **cannot** be used to create objects (in the example above, it is not possible to create an "Animal" object in the MyMainClass)

Interface methods does not have a body - the body is provided by the "implement" class

On implementation of an interface, you must override all of its methods

Interface methods are by default **abstract** and **public**

Interface attributes are by default **public**, **static** and **final**

An interface cannot contain a constructor (as it cannot be used to create objects)

Observations

We have observed that interfaces works kind of like a constructor minus being able to create objects and seems to be used for running methods on their given class/classes. We have yet to figure out how this is smart compared to just using a print command from the main void.

Code words

How to create a Interface. Interface -Name- { Methods to run }

How to connect a interface with a class. Class -Name- implements -Interface name- {
Methods }

Things to look up

Abstract methods seems to be a better and more versatile version of interfaces but also seems to be able to be used in interfaces.

Serialization is a way to make an object into storable data for files or other.

Iterable seems to be a way to run a for-loop on different classes and seems to be at least somewhat similar to interfaces.

Comparable = https://www.youtube.com/watch?time_continue=534&v=swEvHhN9I8k

What is Serialization?

Java provides a mechanism, called “object serialization” where an object can be represented as a sequence of bytes that includes the object's data as well as information about the object's type and the types of data stored in the object.

After a serialized object has been written into a file, it can be read from the file and deserialized that is, the type information and bytes that represent the object and its data can be used to recreate the object in memory.

Classes `ObjectInputStream` and `ObjectOutputStream` are high-level streams that contain the methods for serializing and deserializing an object.

What is java interface iterable?

An interface is at its heart a list of methods that a class should implement. The `iterable` interface is very simple -- there is only one method to implement: `Iterator()`. When a class implements the `Iterable` interface, it is telling other classes that you can get an `Iterator` object to use to iterate over (i.e., traverse) the data in the object.

So basically iterators allow iteration over any `Collection`.

It's also what is required to use Java's for-each control statement.

What is java Comparable and Comparator?

Java `Comparable` interface is used to order the objects of the user-defined class. This interface is found in `java.lang` package and contains only one method named `compareTo(Object)`. It provides a single sorting sequence only, i.e., you can sort the elements on the basis of single data member only. For example, it may be rollno, name, age or anything else.

`CompareTo` is used to compare the current object with the specified object. It can return

1. Positive integer, if the current object is greater than the specified object.
2. Negative integer, if the current object is less than the specified object.
3. Zero, if the current object is equal to the specified object.

Github examples:

Sources:

https://www.w3schools.com/java/java_interface.asp
https://www.tutorialspoint.com/java/java_interfaces.htm
<https://stackoverflow.com/questions/1321122/what-is-an-interface-in-java>
<https://stackoverflow.com/questions/1913098/what-is-the-difference-between-an-interface-and-an-abstract-class/1913185>
<https://examples.javacodegeeks.com/core-java/util/iterator/java-lang-iterable-interface-example/>
https://www.tutorialspoint.com/java/java_serialization.htm
<https://www.javatpoint.com/Comparable-interface-in-collection-framework>
<https://www.youtube.com/watch?v=UumX4mQKQIA&t=1s>
<https://www.youtube.com/watch?v=fX1xNMBTMfg>
<https://www.youtube.com/watch?v=NnZQ-C0x4hs>
<https://www.youtube.com/watch?v=swEvHhN9I8k>
https://www.youtube.com/watch?v=g0_pQCxfgKI
<https://www.youtube.com/watch?v=oAp4GYprVHM>