SYNTAX TECHNOLOGIES

JAVA

Class 26

Agenda

Abstraction in Java

Abstract Class

Abstraction

Abstraction is a process **of hiding the implementation details and showing only functionality to the user**.

For example sending sms, you just type the text and send the message. You don't know the internal processing about the message delivery.

Abstraction is the process of separating ideas from their action.

Abstraction

Real Life Example of Abstraction

Abstraction shows only important things to the user and hides the internal details for example when we use the ATM we don’t care how the balance is being verified from our bank we insert our card enter the details and get the money

Real Life Example of Abstraction

Abstraction

Likewise in Object-oriented programming, **abstraction is a process of hiding the implementation details from the user**, only the functionality will be visible to the user.

Abstraction lets you focus on what the object does instead of how it does it i.e the user will have the information on what the object does instead of how it does it.

Abstraction is nothing but the process of giving an set of guidelines to the user, who are going to implement these guidelines based on their needs.

Abstraction is the concept of exposing only the required essential characteristics and behavior with respect to a context.

Advantage of Abstraction

Abstraction is one of the fundamental principles of Object Oriented Programming languages.

Abstraction helps to reduce the complexity and also improves the maintainability of the system.

Combining abstraction with the concepts of the Encapsulation and Polymorphism, it gives more power to the Object oriented programming languages.

How to achieve Abstraction ?

There are two ways to achieve abstraction in java:

- Abstract class (0 to 100% or partial abstraction)
- Interfaces (0 to 100% or partial abstraction)

Class Type

In java programming we have two types of classes:

Abstract class

Concrete class

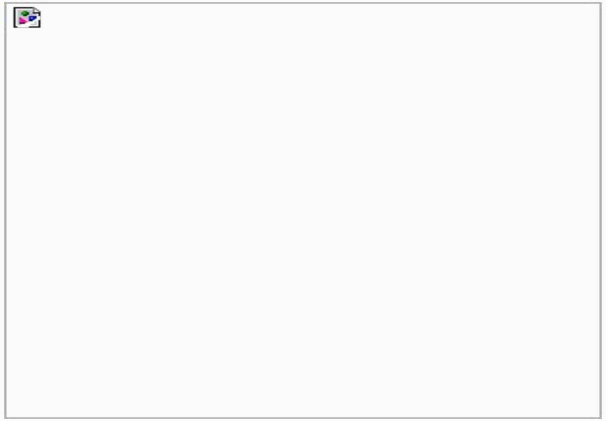
An **abstract class** is one which is declared with abstract keyword and can contain defined methods and undefined method.

Abstract class in Java

1. Class that is declared with abstract keyword, is known as abstract class. An abstract class is one which may contain some defined method and some undefined method.
2. Class is declared abstract, it cannot be instantiated/ cannot create an object of it.
3. In Java programming undefined methods are known as un-implemented or abstract method.
4. To utilize an abstract class, we have to inherit it from another class and provide implementations for the abstract methods in it. (It needs to be extended and its method implemented.)
5. If we inherit an abstract class, we have to provide implementations for all the abstract methods in it.

Abstract class in Java

Syntax:
 	
abstract class
 	
className {
……
}
	Example:
	
	abstract class A {
	…..
	}
		If any class have any abstract method then that class become an abstract class.
		
		Example:
		class Vehicle {
		abstract void Bike();
		}



Abstract Class

An object of abstract class can not be created directly but it can be created indirectly. It means you can create an object of abstract derived class. You can see in above example Example

Vehicle obj = new Bike(); //indirect object creation

If you are extending any abstract class that have abstract method, you must either provide the implementation of the method or make this class abstract.

Make method as abstract method

An abstract method is one which contains only declaration or prototype but it never contains body or definition. In order to make any undefined method as abstract whose declaration is must be predefined by abstract keyword.

Syntax

```
abstract ReturnType methodName(List of formal parameter)
```

Example

```
abstract   void  sum();
abstract  void  diff(int, int);
```

Task

1. Create a class File that will have the following behaviors: open, edit, close. Edit and close are implemented method while open is an abstract. Create 3 subclasses: JavaFile, WordFile, PdfFile that will provide specific implementation of open behaviour: Example: to open .java file we need notepad++ or sublime text, to open .doc file we need Microsoft word to be installed etc

Important Points about abstract class

We use abstract classes when we know the methods, but we don’t know how they would be implemented.

Every abstract class participate in inheritance.

Abstract class of java always contains common features.

Abstract classes definitions should not be made as final because abstract classes always participate in inheritance classes.

An object of abstract class can not be created directly but it can be created indirectly.

Interface in Java

Interface is similar to class which is collection of public static final variables (constants) and public abstract methods.

The interface is also a mechanism to achieve abstraction in java.

There can be only abstract methods in the interface(before java 8) Now we can have default and static methods as well. It is used to achieve fully abstraction and multiple inheritance in Java.

Interface in Java

By using Interface, we can achieve multiple inheritance in java.

It is implicitly abstract. So we don’t need to use the abstract keyword when declaring an interface.

Each method in an interface is also implicitly abstract, so the abstract keyword is not needed.

Methods in an interface are implicitly public.

All the data members of interface are implicitly public static final. Interface can not contain instance fields. Interface only contains public static final variables. The java compiler adds public and abstract keywords before the interface method and public, static and final keywords before data members.

Interface in Java

Interface cannot be extended by a class; it is implemented by a class.

Interface can extend multiple interfaces. It means interface support multiple inheritance

An interface reference can point to objects of its implementing classes

An interface is not a class. Writing an interface is similar to writing a class, but they are two different concepts. A class describes the attributes and behaviors of an object. An interface contains behaviors that a class implements.

Unless the class that implements the interface is abstract, all the methods of the interface need to be defined in the class.