

Assignment 2

Question: WAP to show all the differences between abstract class and interface covering all the access specifiers you learn in last class.

Answer: Here's a program in Java that demonstrates the differences between abstract classes and interfaces, covering all access specifiers:

```
// abstract class
abstract class AbstractClass {

    // abstract method with public access specifier
    public abstract void publicMethod();

    // abstract method with protected access specifier
    protected abstract void protectedMethod();

    // abstract method with default access specifier
    abstract void defaultMethod();

    // abstract method with private access specifier
    // (this is not possible in an interface)
    private abstract void privateMethod();

    // concrete method with public access specifier
    public void concretePublicMethod() {
        // implementation
    }

    // concrete method with protected access specifier
    protected void concreteProtectedMethod() {
        // implementation
    }
}
```

```
// concrete method with default access specifier
void concreteDefaultMethod() {
    // implementation
}

// concrete method with private access specifier
private void concretePrivateMethod() {
    // implementation
}

// interface
interface MyInterface {

    // abstract method with public access specifier
    public void publicMethod();

    // abstract method with default access specifier
    void defaultMethod();

    // (there is no way to declare an abstract method with protected or
    private access specifier in an interface)

    // default method with public access specifier
    public default void defaultPublicMethod() {
        // implementation
    }

    // default method with default access specifier
    default void defaultDefaultMethod() {
        // implementation
    }
}
```

```
}
```

```
// static method with public access specifier
```

```
public static void staticPublicMethod() {
```

```
    // implementation
```

```
}
```

```
// static method with default access specifier
```

```
static void staticDefaultMethod() {
```

```
    // implementation
```

```
}
```

```
}
```

```
// implementation of abstract class
```

```
class ConcreteClass extends AbstractClass {
```

```
// implementation of public abstract method
```

```
public void publicMethod() {
```

```
    // implementation
```

```
}
```

```
// implementation of protected abstract method
```

```
protected void protectedMethod() {
```

```
    // implementation
```

```
}
```

```
// implementation of default abstract method
```

```
void defaultMethod() {
```

```
    // implementation
```

```
}
```

```
// implementation of private abstract method
```

```
private void privateMethod() {  
    // implementation  
}  
  
// implementation of interface  
class MyClass implements MyInterface {  
  
    // implementation of public abstract method  
    public void publicMethod() {  
        // implementation  
    }  
  
    // implementation of default abstract method  
    public void defaultMethod() {  
        // implementation  
    }  
}
```

The differences between abstract classes and interfaces with respect to access specifiers are as follows:

Abstract classes can have abstract methods with any access specifier (public, protected, default, or private), while interfaces can only have abstract methods with public or default access specifier.

Abstract classes can have concrete methods with any access specifier (public, protected, default, or private), while interfaces can only have default or static methods with any access specifier, or public default methods.

Abstract classes can be subclassed by a concrete class, while interfaces can be implemented by a class.