Java

Create Object in 5 Ways

@vikasrajputin

1. Using a new keyword.

➔ It's the most popular one. We create an object by using a new operator followed by a constructor call.

➔ Example:

```
class ClassA{

}

ClassA obj=new ClassA( );
```

@vikasrajputin

swipe left

2. newInstance() method

➔ Using the newInstance() method of class "Class"

➔ Example:

```
class ClassA{

}

ClassA obj = ClassA.class.newInstance( );
```

3. newInstance() method - 2

➔ Using the newInstance() method in class "Constructor":

➔ Example:

```
class ClassA{

}

Constructor<ClassA> obj =
  ClassA.class.getConstructor().newInstance();
```

➔ Both the previous ways (Shown in 2 and 3), are known as reflective ways of creating objects.

Fun-fact:

➔ Class's newInstance() method internally uses Constructor's newInstance() method.

4. clone() method

➔ Using "Object" class clone() method. The clone() method creates a copy of an existing object.

➔ The clone() method is part of the "Object" class which returns a clone object.

➔ Example:

```
public class ClassA implements Cloneable {
  protected Object clone() throws CloneNotSupportedException {
    //...
  }
}

ClassA obj1 = new ClassA();
ClassA obj2 = (ClassA) obj1.clone();
```

➔ When using the clone() method:

➔ **Always Remember:**

- The "Cloneable" interface is implemented.

- The clone() method must be overridden with other classes.

- Inside the clone() method, the class must call super.clone().

5. Deserialization

- ➔ When we deserialize any object then JVM creates a new object internally.

- ➔ For this, we need to implement the Serializable interface.

- ➔ Example:

```
public class ClassA implements Serializable {
    //...
}

//  Serialization
ClassA classA;
try (ObjectOutputStream out = new ObjectOutputStream(
  new FileOutputStream("classA.obj"))) {
    out.writeObject(classA);
}

//  Deserialization
ClassA deserialClassA;
try (ObjectInputStream in = new ObjectInputStream(
  new FileInputStream("classA.obj"))) {
    deserialClassA = (ClassA) in.readObject();
}

// deserialClassA Object will be created after de-
serialization process
```

❤️ **Thanks for reading !**

For more content on

Java & Backend Development,

follow me on below handles

Vikas Rajput

@vikasrajputin