JAVA BEANS

- **Definition**: Java bean is a reusable software component that can be manipulated visually in a builder tool.
- Java bean is an object of java which is serializable with a nullary constructor i.e. a constructor with no arguments and allow access to the properties using getter and setter methods.
- **Advantages of Java beans**:
  - Java Bean obtains all the benefits of Java's "write-once, run-anywhere" paradigm (reusability).
  - The properties, events, and methods of a Java Bean that are exposed to an application builder tool can be controlled.
  - The configuration settings of a Java Bean can be saved in persistent storage and restored at a later time.
  - Java bean has the capability to receive the events from an object and also generate the events to send to another object.
- **BDK Introspection**:
  - It is the process of analyzing a bean to determine it’s capabilities.
  - It is a very important feature of Java Bean API because it allows an application builder to present the information about a component to the software designer.

  - Without introspection java bean is difficult to operate.

- One way to expose the properties, events and methods of java bean to application builder tool is using simple naming conventions.
- **Properties**: Property is a subset of bean’s state. The values that are assigned to the properties determine the behavior and appearance of the component.
- There are different types of properties:
- Simple Property: Simple property has only single value . It can be represented by the following example where N is the name of the property and T is the type of the property.

  Public T getN();

  Public void setN();
- Boolean Property: Boolean property has the value of true or false. It can be represented by the following example where N is the name of the property.

  Public Boolean isN();

  Public Boolean getN();

  Public void setN(Boolean value);
- Indexed Property: it consists of multiple values. It can be represented by the following example where N is the name of the property and T is its type.

```
Public T getN(int index);
Public void setN(int index, T value);
Public T[ ] getN();
Public void setN(T values[ ]);
```

- Using Bound Property: A bean that has a bound property generates an event when the property is changed. The event is of type **PropertyChangeEvent** and is sent to objects that are previously registered to receive such notifications.
- *Example:* for bound property is the applications that uses the tick tock bean to automatically control the color bean.
- **Using BeanInfo interface**:
- It defines several methods like :

```
PropertyDescriptor[ ]    getPropertyDescriptors( )
EventSetDescriptor[  ]   getEventSetDescriptors( )
MethodDescriptor[ ]      getMethodDescriptors( )
```

- The above methods returns array of objects that provides the information about properties, events and methods of a bean.
- SimpleBeanInfo is a class that provides the default implementations of BeanInfo interface including the above three methods.

- **Constrained Properties**: A bean that has a constrained property will generates an event when an attempt is made to change its value. The event is of type **PropertyChangeEvent** and is sent to objects that are previously registered to receive such notifications.
- This capability allows a Bean to operate differently according to its run-time environment.
- **Persistence**: Persistence is the ability to save a Bean to non volatile storage and retrieve it at a later time. The information that is particularly important are the configuration settings.
- **Customizers**: A bean developer can provide a customizer to help another developer to configure this software. A customizer can provide a step-by-step procedure through the process that must be followed to use the component in a specific context.

Java Beans API

- Java bean API consists of classes and interfaces

Interface	Description
AppletInitializer	Methods in this interface are used to initialize Beans that are also applets.
BeanInfo	It allows the designer to specify information about properties, events & methods of a Bean.
Customizer	It allows the designer to provide a graphical user interface through which a Bean may be configured.
DesignMode	Used to determine whether a Bean is executing in design mode or not.
PropertyChangeListener	It is invoked when a bound property is changed.
Visibility Methods	It allows a bean to execute in the environments where graphical user interface is not available.

Class	Description
BeanDescriptor	It is a class that provides the information about a bean.
Beans	It is a class used to obtain the information about a bean.
IntrospectionException	It is generated if a problem occurs while analyzing the bean.
PropertyChangeEvent	It is generated when bound or constrained properties are changed.
PropertyDescriptor	Instance of this class describes the property of a bean.