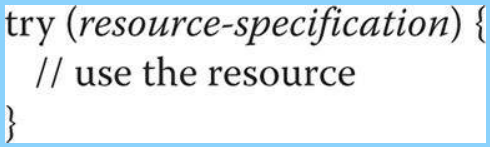
TOPIC:

- TRY WITH RESOURCES
- EVENT HANDLING

TRY WITH RESOURCES

•A feature that offers another way to manage resources, such as file streams, by automating the closing process.
•Also referred as *automatic resource management, or ARM.*
•Principal advantage : prevents situations in which a file (or other resource) is inadvertently not released after it is no longer needed.
•General form:



```
try (resource-specification) {
   // use the resource
}
```

•*resource-specification: declares and initializes a* resource, such as a file stream.
•When the try block ends, the resource is automatically released.
•No need to call close( ) explicitly.
•this form of try can also include catch and finally clauses.

do example given on page 482-483 of pdf,10 edition

EVENT HANDLING

•Any program that uses a graphical user interface, such as a Java application written for Windows, is event driven.
•Supported by a number of packages, including java.util, java.awt, and java.awt.event.

The Delegation Event Model
•Concept : a source generates an event and sends it to one or more listeners.
•listener simply waits until it receives an event. Once an event is received, the listener processes the event and then returns.
•listeners must register with a source in order to receive an event notification.
•notifications are sent only to listeners that want to receive them.

EVENTS

•An event is an object that describes a state change in a source.
•Some of the activities that cause events to be generated are pressing a button, entering a character via the keyboard, selecting an item in a list, and clicking the mouse.

Event Sources
•A source is an object that generates an event.
•A source must register listeners in order for the listeners to receive notifications about a specific type of event.
• Each type of event has its own registration method.
•General form: public void addTypeListener (TypeListener el)
•Here, Type is the name of the event, and el is a reference to the event listener.
•For example, the method that registers a keyboard event listener is called addKeyListener( ).

Event Sources

•A source must also provide a method that allows a listener to unregister an interest in a specific type of event.

•The general form: public void removeTypeListener(TypeListener el)

•Here, Type is the name of the event, and el is a reference to the event listener.

• For example, to remove a keyboard listener, you would call removeKeyListener( ).

•The methods that add or remove listeners are provided by the source that generates events.

•For example, Component, which is a top-level class defined by the

•AWT, provides methods to add and remove keyboard and mouse event listeners.

Event Listeners

•A listener is an object that is notified when an event occurs.

•It has two major requirements. First, it must have been registered with one or more sources to receive notifications about specific types of events. Second, it must implement methods to receive and process these notifications.

• In other words, the listener must supply the event handlers.

•The methods that receive and process events are defined in a set of interfaces, such as those found in java.awt.event.

•For example, the MouseMotionListener interface defines two methods to receive notifications when the mouse is dragged or moved. Any object may handle one or both of these events if it provides an implementation of this interface.

Event Classes

•The classes that represent events are at the core of Java’s event handling mechanism.

•At the root of the Java event class hierarchy is EventObject, which is in java.util. It is the superclass for all events.

•Its one constructor is shown here: EventObject(Object src)

•Here, src is the object that generates this event.

•EventObject defines two methods: getSource( ) and toString( ).

•The getSource() method returns the source of the event.

• Its general form is shown here: Object getSource( )

• toString( ) returns the string equivalent of the event.

•The class AWTEvent, defined within the java.awt package, is a subclass of EventObject. Its getID( ) method can be used to determine the type of the event.

•The signature of this method is shown here: int getID( )

The ActionEvent Class

•An ActionEvent is generated when a button is pressed, a list item is double-clicked, or a menu item is selected.

•The ActionEvent class defines four integer constants: ALT_MASK,

•CTRL_MASK, META_MASK, and SHIFT_MASK.

•ActionEvent has these three constructors:

```
ActionEvent(Object src, int type, String cmd)
ActionEvent(Object src, int type, String cmd, int modifiers)
ActionEvent(Object src, int type, String cmd, long when, int
modifiers)
```

•Here, src is a reference to the object that generated this event.

•The type of the event is specified by type, and its command string is cmd.

•The argument modifiers indicates which modifier keys (alt, ctrl, meta, and/or shift) were pressed.

•when parameter specifies when the event occurred.

References/Resources:

- Balaguruswamy, E. (2014). Programming with JAVA: A Primer. 5th edition. India: McGraw Hill Education
- Horstmann, C. S. (2017). Core Java - Vol. I – Fundamentals (Vol. 10). Pearson Education
- Schildt, H. (2018). Java: The Complete Reference. 10th edition. McGraw-Hill Education.

NOTE: Please go through the reference book for details on the above topic and feel free to mail your doubts or discuss anything.

Assignment

Q1. Write a program that copies content of one file to another. Pass the names of the files through command-line arguments. (Try on machine also, if possible)

Q2. Write a program in Java (using try-with-resources functionality) to do the following:

i)open two files "exam1.txt" and "exam2.txt". Accept file names through command-line arguments.

ii) exit the program if any of the two files is unable to open.

iii) append contents of "exam1.txt" to "exam2.txt".

iv) re-write contents of "exam2.txt" after removing all white spaces from the updated content (without using built-in methods).

Assignment

Q3. What is Delegation Event Model? Explain its components in short.

Q4. Write commonly used Event Classes in java.awt.event and their description.