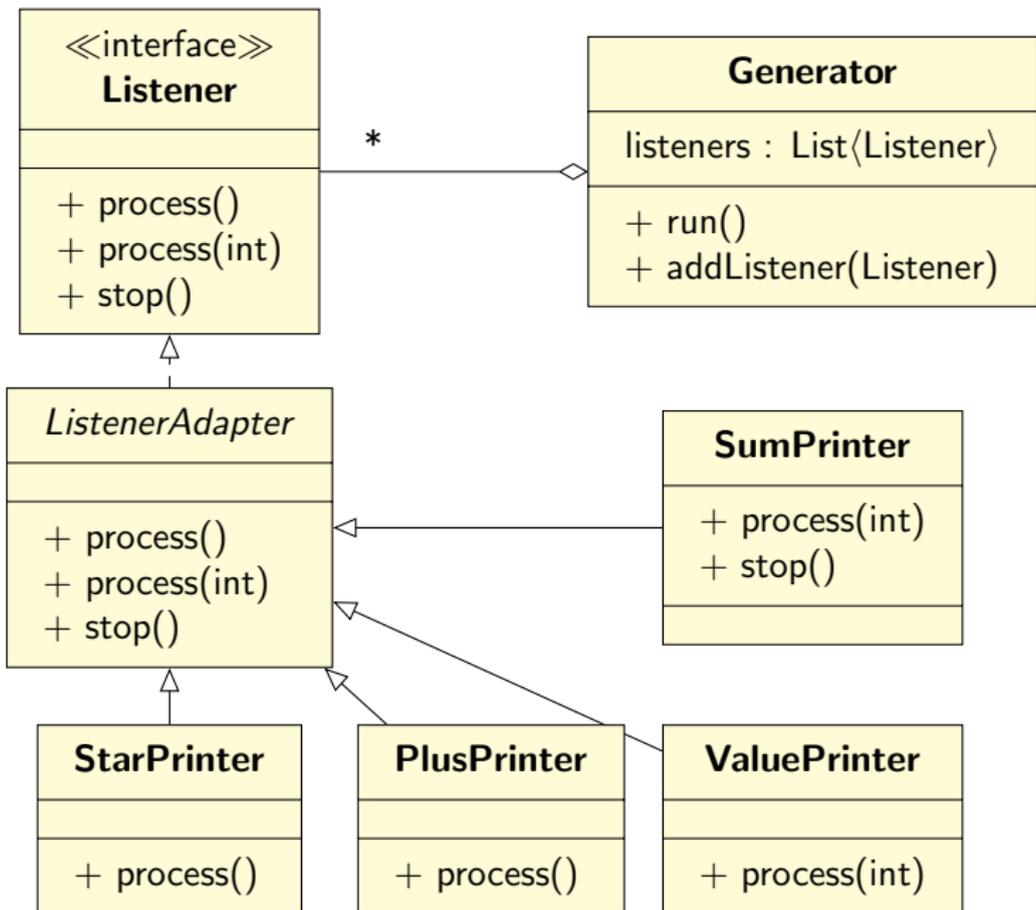


Listen
EECS 4315

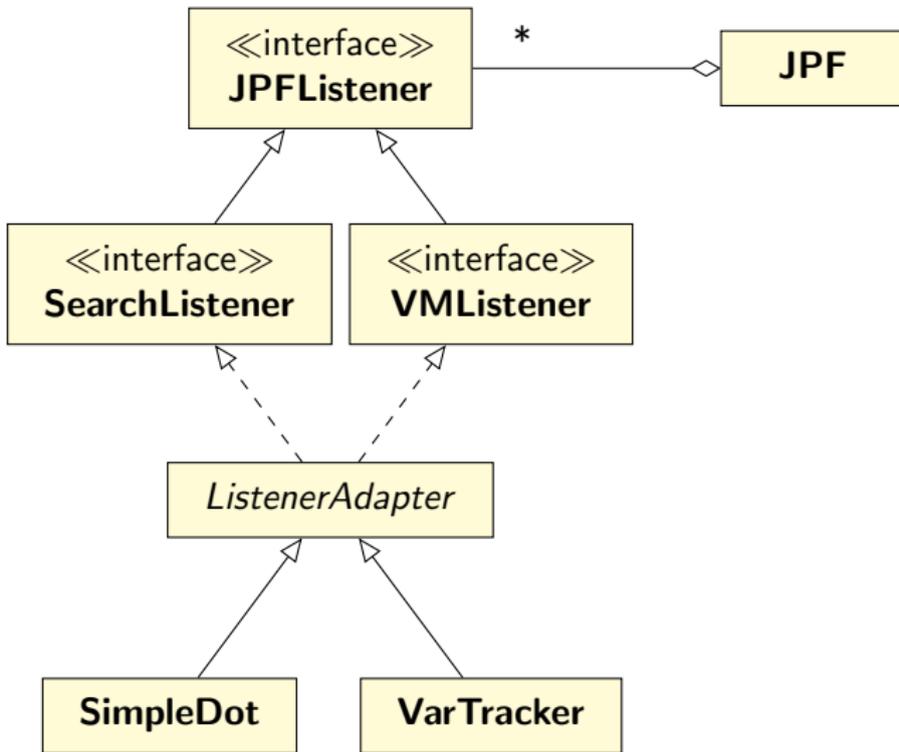
www.eecs.yorku.ca/course/4315/



Generator and listeners



JPF and listeners



The interface `JPFListener` is empty.

The interface `JPFListener` is empty.

Question

Why introduce an empty interface?

The interface `JPFListener` is empty.

Question

Why introduce an empty interface?

Answer

JPF has a collection of `JPFListeners`, some can be `SearchListeners` and others can be `VMListeners`.

SearchListener

```
public interface SearchListener extends JPFLListener {
    void stateAdvanced(Search search);
    void stateBacktracked(Search search);
    void stateProcessed(Search search);
    void statePurged(Search search);
    void stateRestored(Search search);
    void stateStored(Search search);

    void searchProbed(Search search);

    void propertyViolated(Search search);

    void searchConstraintHit(Search search);

    void searchStarted(Search search);
    void searchFinished(Search search);
}
```

Implement a listener which prints the states and transitions visited by the search in the following simple format:

0 -> 1

1 -> 2

0 -> 3

3 -> 4

4 -> 2

Question

Which methods of the `SearchListener` interface are relevant?

Question

Which methods of the `SearchListener` interface are relevant?

Answer

`stateAdvanced`, `stateBacktracked`, and `stateRestored`.

Question

In order to print a transition, what information do we need?

Question

In order to print a transition, what information do we need?

Answer

The ID of the source and target state.

State space

Question

In order to print a transition, what information do we need?

Answer

The ID of the source and target state.

Question

How do we store that information?

State space

Question

In order to print a transition, what information do we need?

Answer

The ID of the source and target state.

Question

How do we store that information?

Answer

As attributes.

State space

Question

In order to print a transition, what information do we need?

Answer

The ID of the source and target state.

Question

How do we store that information?

Answer

As attributes.

```
private int previous;  
private int current;
```

```
public void stateAdvanced(Search search) {  
    this.previous = ???;  
    this.current = ???;  
}
```

Question

How do we update `this.previous`?

```
public void stateAdvanced(Search search) {  
    this.previous = ???;  
    this.current = ???;  
}
```

Question

How do we update `this.previous`?

Answer

```
this.previous = this.current.
```

```
public void stateAdvanced(Search search) {  
    this.previous = ???;  
    this.current = ???;  
}
```

Question

How can we use the `Search` parameter of the `stateAdvanced` method to update `this.current`?

```
public void stateAdvanced(Search search) {  
    this.previous = ???;  
    this.current = ???;  
}
```

Question

How can we use the `Search` parameter of the `stateAdvanced` method to update `this.current`?

Answer

Use a method of the `Search` class that returns the ID of the current state (`getStateId`).

Question

Where do we initialize the attributes `current` and `previous`?

Question

Where do we initialize the attributes `current` and `previous`?

Answer

In the constructor.

Question

Where do we initialize the attributes `current` and `previous`?

Answer

In the constructor.

Question

How do we initialize the attributes `current` and `previous`?

Question

Where do we initialize the attributes `current` and `previous`?

Answer

In the constructor.

Question

How do we initialize the attributes `current` and `previous`?

Answer

Set them to -1 , the ID of the initial state.

Question

How do we print the transition in `stateAdvanced`?

Question

How do we print the transition in `stateAdvanced`?

Answer

```
System.out.printf("%d -> %d\n", this.previous, this.current);
```

Question

How do we implement `stateBacktracked`?

Question

How do we implement `stateBacktracked?`

Answer

```
this.current = search.getStateId();
```

Question

How do we implement `stateRestored`?

Question

How do we implement `stateRestored`?

Answer

```
this.current = search.getStateId();
```

To compile the listener, make sure that `jpf.jar` is part of the classpath.

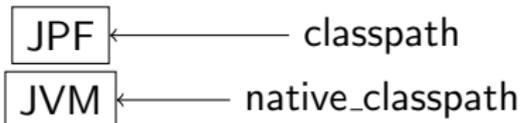


- JPF is a JVM.
- Since JPF is written in Java, it runs on a JVM.

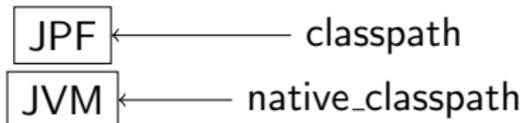


- JPF is a JVM.
- Since JPF is written in Java, it runs on a JVM.
- JPF model checks Java bytecode.
- JVM executes Java bytecode.

Each JVM has a classpath which tells the JVM where to look for classes.



Each JVM has a classpath which tells the JVM where to look for classes.



classpath of JPF: where JPF looks for classes to model check

native_classpath of JPF: where the JVM looks for classes to execute (as part of JPF)

Implement a listener which creates a dot file representing the the states and transitions visited by the search.

```
digraph statespace {  
0 -> 1  
1 -> 2  
0 -> 3  
3 -> 4  
4 -> 2  
}
```

Question

Where do we open a file for writing?

Question

Where do we open a file for writing?

Answer

In the constructor.

Question

Where do we print `digraph statespace` {?

Question

Where do we print `digraph statespace` {?

Answer

In the method `searchStarted`.

Question

Where do we print `digraph statespace {`?

Answer

In the method `searchStarted`.

Question

Where do we print the final `}`?

Question

Where do we print `digraph statespace {?`

Answer

In the method `searchStarted`.

Question

Where do we print the final `}?`

Answer

In the method `searchFinished`.

Implement a listener which creates a dot file representing the the states and transitions visited by the search. Colour the initial state green and the final states red.

```
digraph statespace {  
0 [fillcolor=green]  
0 -> 1  
1 -> 2  
2 [fillcolor=red]  
0 -> 3  
3 -> 4  
4 -> 2  
}
```

Question

The initial state always has ID 0. Where do we print

```
0 [fillcolor=green]?
```

Question

The initial state always has ID 0. Where do we print
0 [fillcolor=green]?

Answer

In the method `searchStarted`.

Question

The class `Search` has a method `isEndState`. How can this method be used?

Question

The class `Search` has a method `isEndState`. How can this method be used?

Answer

To indicate that the final (end) states are red.