CREATIVE ASSIGNMENT

On

"ITERATOR DESIGN PATTERN"

Submitted By

Mr. Pushpak Fasate

Under the Guidance of

Ms. Titiksha Bhagat

Department of Computer Science & Engineering

S. B. Jain Institute of Technology Management and Research, Nagpur-441501
2020-2021

Suppose we have a list of Radio channels and the client program want to traverse through them one by one or based on the type of channel. For example some client programs are only interested in English channels and want to process only them, they don't want to process other types of channels.

Code :-

1) <u>Filename :- ChannelTypeEnum.java</u>

```
public enum ChannelTypeEnum {
        ENGLISH, HINDI, FRENCH, ALL;
}
```

2) <u>Filename :- Channel.java</u>

```
public class Channel {

        private double frequency;
        private ChannelTypeEnum TYPE;

        public Channel(double freq, ChannelTypeEnum type){
                this.frequency=freq;
                this.TYPE=type;
        }

        public double getFrequency() {
                return frequency;
        }

        public ChannelTypeEnum getTYPE() {
                return TYPE;
        }

        @Override
        public String toString(){
                return "Frequency="+this.frequency+", Type="+this.TYPE;
        }
}
```

3) <u>Filename :- ChannelCollection.java</u>

```
public interface ChannelCollection {
        public void addChannel(Channel c);
        public void removeChannel(Channel c);
```

```
	public ChannelIterator iterator(ChannelTypeEnum type);
}
```

4) <u>Filename :- ChannelIterator.java</u>

```
public interface ChannelIterator {
	public boolean hasNext();
	public Channel next();
}
```

5) <u>Filename :- ChannelCollectionImpl.java</u>

```
import java.util.ArrayList;
import java.util.List;

public class ChannelCollectionImpl implements ChannelCollection {

	private List<Channel> channelsList;

	public ChannelCollectionImpl() {
		channelsList = new ArrayList<>();
	}

	public void addChannel(Channel c) {
		this.channelsList.add(c);
	}

	public void removeChannel(Channel c) {
		this.channelsList.remove(c);
	}

	@Override
	public ChannelIterator iterator(ChannelTypeEnum type) {
		return new ChannelIteratorImpl(type, this.channelsList);
	}

	private class ChannelIteratorImpl implements ChannelIterator {

		private ChannelTypeEnum type;
		private List<Channel> channels;
		private int position;
		public ChannelIteratorImpl(ChannelTypeEnum ty,
				List<Channel> channelsList) {
			this.type = ty;
			this.channels = channelsList;
		}

		@Override
		public boolean hasNext() {
			while (position < channels.size()) {
				Channel c = channels.get(position);
				if (c.getTYPE().equals(type) || type.equals(ChannelTypeEnum.ALL)) {
					return true;
				} else
					position++;
			}
			return false;
		}

		@Override
		public Channel next() {
			Channel c = channels.get(position);
			position++;
			return c;
		}

	}
}
```

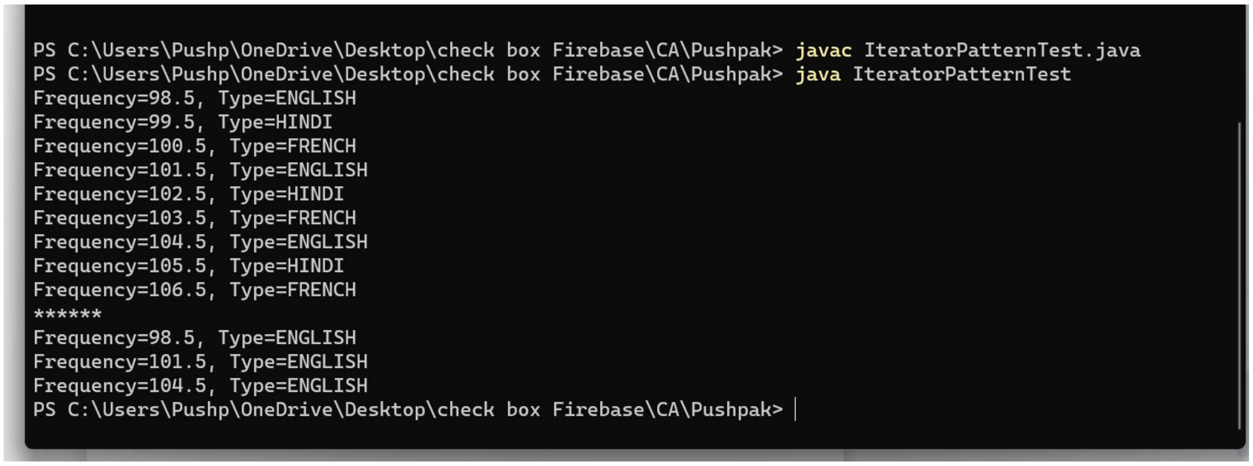
6) <u>Filename :- IteratorPatternTest .java</u>

```
public class IteratorPatternTest {

	public static void main(String[] args) {
		ChannelCollection channels = populateChannels();
		ChannelIterator baseIterator = channels.iterator(ChannelTypeEnum.ALL);
		while (baseIterator.hasNext()) {
			Channel c = baseIterator.next();
			System.out.println(c.toString());
		}
		System.out.println("******");
		// Channel Type Iterator
		ChannelIterator englishIterator = channels.iterator(ChannelTypeEnum.ENGLISH);
		while (englishIterator.hasNext()) {
			Channel c = englishIterator.next();
			System.out.println(c.toString());
		}
	}

	private static ChannelCollection populateChannels() {
		ChannelCollection channels = new ChannelCollectionImpl();
		channels.addChannel(new Channel(98.5, ChannelTypeEnum.ENGLISH));
		channels.addChannel(new Channel(99.5, ChannelTypeEnum.HINDI));
		channels.addChannel(new Channel(100.5, ChannelTypeEnum.FRENCH));
		channels.addChannel(new Channel(101.5, ChannelTypeEnum.ENGLISH));
		channels.addChannel(new Channel(102.5, ChannelTypeEnum.HINDI));
		channels.addChannel(new Channel(103.5, ChannelTypeEnum.FRENCH));
		channels.addChannel(new Channel(104.5, ChannelTypeEnum.ENGLISH));
		channels.addChannel(new Channel(105.5, ChannelTypeEnum.HINDI));
		channels.addChannel(new Channel(106.5, ChannelTypeEnum.FRENCH));
		return channels;
	}
}
```

<u>Output</u> : -



```
PS C:\Users\Pushp\OneDrive\Desktop\check box Firebase\CA\Pushpak> javac IteratorPatternTest.java
PS C:\Users\Pushp\OneDrive\Desktop\check box Firebase\CA\Pushpak> java IteratorPatternTest
Frequency=98.5, Type=ENGLISH
Frequency=99.5, Type=HINDI
Frequency=100.5, Type=FRENCH
Frequency=101.5, Type=ENGLISH
Frequency=102.5, Type=HINDI
Frequency=103.5, Type=FRENCH
Frequency=104.5, Type=ENGLISH
Frequency=105.5, Type=HINDI
Frequency=106.5, Type=FRENCH
******
Frequency=98.5, Type=ENGLISH
Frequency=101.5, Type=ENGLISH
Frequency=104.5, Type=ENGLISH
PS C:\Users\Pushp\OneDrive\Desktop\check box Firebase\CA\Pushpak>
```