Experiment – 12

Aim –Iterator design pattern (Behavioral pattern)

Concept -client does not know the internal data structure of concrete containers instead it uses iterator.

iIterator.java

```
public interface iIterator {
    public boolean hasnext();
    public Object next();
}
```

iContainer.java

```
public interface iContainer {
    public iIterator getIterator();
}
```

FoodContainer.java

```
public class FoodContainer implements iContainer{
    private String food[] = {"burger","pizza","sandwich","dhokla"};

    @Override
    public iIterator getIterator() {
        return new FoodIterator();
    }

    private class FoodIterator implements iIterator{
        private int index;

        @Override
        public boolean hasnext() {
            if (index < food.length){
                return true;
            }else {
                return false;
            }
        }

        @Override
        public Object next() {
            if (this.hasnext()){
                return food[index++];
            }else {
                return null;
            }
```

```
        }
    }
}
```

Client.java

```
public class Client {

    public static void main(String[] args) {

        iContainer container = new FoodContainer();
        iIterator iterator = container.getIterator();
        while (iterator.hasnext()){
            Object obj = iterator.next();
            System.out.println(obj);
        }

    }
}
```

OUTPUT :

```
"C:\Program Files\Java\jdk-17.0.2\bin\java.exe" "-javaagent:C:\Progra
burger
pizza
sandwich
dhokla

Process finished with exit code 0
```