ICS 372 Object-Oriented Design and Implementation
Class Exercise 4 Solution

Q17. Object Adapter

```
import java.util.ArrayList;
import java.util.Iterator;

public class DequeImplementationObjectAdapter implements Deque {
	private ArrayList list = new ArrayList();

	@Override
	public boolean addAtTail(Object value) {
		list.add(value);
		return true;
	}

	@Override
	public Object removeElementAtTail() {
		if (list.size() > 0) {
			return list.remove(list.size() - 1);
		}
		return null;
	}

	@Override
	public Object getElementAtTail() {
		if (list.size() > 0) {
			return list.get(list.size() - 1);
		}
		return null;
	}

	@Override
	public boolean addAtHead(Object value) {
		list.add(0, value);
		return true;
	}

	@Override
	public Object removeElementAtHead() {
		if (list.size() > 0) {
			return list.remove(0);
		}
		return null;
	}

	@Override
	public Object getElementAtHead() {
		if (list.size() > 0) {
			return list.get(0);
		}
		return null;
	}

	@Override
	public int size() {
```

```
                return list.size();
        }

        @Override
        public void clear() {
                list.clear();
        }

        @Override
        public Iterator iterator() {
                return list.iterator();
        }

}
```

Q18. Class Adapter

```
import java.util.ArrayList;

public class DequeImplementationClassAdapter extends ArrayList implements Deque {
        @Override
        public boolean addAtTail(Object value) {
                super.add(value);
                return true;
        }

        @Override
        public Object removeElementAtTail() {
                if (size() > 0) {
                        return super.remove(size() - 1);
                }
                return null;
        }

        @Override
        public Object getElementAtTail() {
                if (size() > 0) {
                        return super.get(super.size() - 1);
                }
                return null;
        }

        @Override
        public boolean addAtHead(Object value) {
                super.add(0, value);
                return true;
        }

        @Override
        public Object removeElementAtHead() {
                if (super.size() > 0) {
                        return super.remove(0);
                }
                return null;
        }

        @Override
        public Object getElementAtHead() {
                if (size() > 0) {
```

```
            return super.get(0);
        }
        return null;
    }

}
```

A Short Test Program

```
import java.util.Iterator;

public class Driver {
    public static void main(String[] args) {
        Deque deque = new DequeImplementationObjectAdapter();
        deque.addAtHead("E1");
        deque.addAtTail("E2");
        deque.addAtTail("E3");
        assert deque.getElementAtHead().equals("E1");
        assert deque.getElementAtTail().equals("E3");
        assert deque.size() == 3;
        deque.addAtTail("E4");
        assert deque.removeElementAtHead().equals("E1");
        assert deque.removeElementAtTail().equals("E4");
        deque.clear();
        assert deque.size() == 0;
        deque.addAtTail("E1");
        deque.addAtTail("E2");
        deque.addAtTail("E3");
        deque.addAtTail("E4");
        Iterator iterator = deque.iterator();
        int count = 1;
        while (iterator.hasNext()) {
            assert iterator.next().equals("E" + count);
            count++;
        }
    }
}
```