```
package simcode;
import javasim.Entity;
import javasim.Queue;
import javasim.timeTable;

/**
 * @author Michael Kepple
 * @version 22 Mar 2013
 * @see JavaSim, Streams
 */
public class SingleTeller
{
        static final int SECINHOUR = 3600;
        static final int SERVICETIME = 45;
        static final int BEGINSIM = 0;
        static final int ARRIVAL = 1;
        static final int DEPART = 2;
        static final int ENDSIM = 3;

        public static void main(String[] args)
        {
                new SingleTeller();
        }

        public SingleTeller()
        {
                boolean run = true;
                Entity endShift, startShift, retrieved;
                Queue waitingCustomers = new Queue();
                timeTable table = new timeTable();
                int clock = 0, entityId = 0, eventRetrieved = 0, customersServed =
0;
                startShift = new Entity(entityId, "Start of Shift");
                startShift.event = 0;
                while (clock < SECINHOUR)
                {
                        clock += 60;
                        entityId++;
                        Entity custArrive = new Entity(entityId, "Customer Arrival"
);
                        custArrive.event = 1;
                        table.setim(custArrive, clock);
                }
                endShift = new Entity(entityId, "End of Shift");
                endShift.event = ENDSIM;
                table.setim(endShift, clock);
                while (run)
                {
                        retrieved = table.scan();
                        eventRetrieved = retrieved.event;
                        switch (eventRetrieved)
                        {
                        case BEGINSIM:
                                System.out.println("Bank simulation has begun");
                                break;
                        case ARRIVAL:
                                System.out.println("Customer arrived at: " + retrie
ved.time);
                                waitingCustomers.addTo(retrieved);
                                // Set their departure
                                retrieved.event = 2;
                                table.setim(retrieved, (retrieved.time+SERVICETIME)
);
```

```
                                break;
                            case DEPART:
                                waitingCustomers.remove(retrieved);
                                customersServed++;
                                System.out.println("Customer left at: " + retrieved
    .time);
                                break;
                            case ENDSIM:
                                System.out.println("Hour is up - Simulation has end
    ed");
                                waitingCustomers.clear();
                                run = false;
                            }
                        }
                        System.out.println("Customers Served: " + customersServed);
                        System.out.println("Minimum Service Time: " + SERVICETIME);
                        System.out.println("Maximum Service Time: " + SERVICETIME);
                        System.out.println("Average Service Time: " + SERVICETIME);
                }

            }
```