

```
package voyage;

public interface TrajetRoutier extends Voyage {
}
```

```
package voyage;

public class TrajetRoutierFactory extends AbstractVoyageFactory {

    public static TrajetRoutierFactory instance;

    private TrajetRoutierFactory () {
    }

    public TrajetRoutier create() {
        return new TrajetRoutierImpl();
    }

    public static TrajetRoutierFactory getInstance(){
        if (instance == null)
            instance = new
TrajetRoutierFactory();
        return instance;
    }
}
```

```
package voyage;

public class VolImpl implements Vol {

    private String destination;
    private String depart;

    public VolImpl() {
    }

    public VolImpl(String dep, String des) {
        this.depart = dep;
        this.destination = des;
    }

    public String getDepart() {
        return depart;
    }

    public void setDepart(String dep) {
        depart = dep;
    }

    public String getDestination() {
        return destination;
    }

    public void setDestination(String des) {
        destination = des;
    }
}
```

```
package voyage;

public interface Vol extends Voyage {
}
```

```
package voyage;

public class TrajetRoutierImpl implements TrajetRoutier {

    private String destination;
    private String depart;

    public TrajetRoutierImpl() {
    }

    public TrajetRoutierImpl(String dep, String des) {
        this.depart = dep;
        this.destination = des;
    }

    public String getDepart() {
        return depart;
    }

    public void setDepart(String dep) {
        depart = dep;
    }

    public String getDestination() {
        return destination;
    }

    public void setDestination(String des) {
        destination = des;
    }
}
```

```
package voyage;

public class VolFactory extends AbstractVoyageFactory {
    private static VolFactory instance;

    private VolFactory() {
    }

    public Vol create() {
        return new VolImpl();
    }

    public static VolFactory getInstance(){
        if (instance == null)
            instance = new VolFactory();
        return instance;
    }
}
```

```
package voyage;

public interface Voyage {

    public String getDepart();

    public void setDepart(String dep);

    public String getDestination();

    public void setDestination(String des);
}
```

```

package reservation;

public class Reservation {

    private int identifiant;
    private Voyage voyage;
    private Passager passager;

    public Passager getPassager() {
        return passager;
    }

    public void setPassager(Passager passager) {
        this.passager = passager;
    }

    public Voyage getVoyage() {
        return voyage;
    }

    public void setVoyage(Voyage voyage) {
        this.voyage = voyage;
    }

    public int getIdentifiant() {
        return this.identifiant;
    }

    public void setIdentifiant(int val) {
        this.identifiant = val;
    }

    public void affiche() {
        System.out.println("Reservation : " + identifiant);
        System.out.println("Passager : " + passager.getNom());
        System.out.println("Voyage : " + voyage.getDepart() + "-" +
voyage.getDestination());
    }
}

```

```

package reservation;

import voyage.TrajetRoutier;
import voyage.Vol;

public class ReservationBuilder extends
AbstractReservationBuilder {

    private static ReservationBuilder instance;

    public void buildReservationVol(Vol v, Passager p) {
        reservation.setVoyage(v);
        reservation.setPassager(p);
    }

    public void buildReservationTrajetRoutier(TrajetRoutier t,
Passager p) {
        reservation.setVoyage(t);
        reservation.setPassager(p);
    }

    public static ReservationBuilder getInstance() {
        if (instance == null) {
            instance = new ReservationBuilder();
        }
        return instance;
    }
}

```

```

package reservation;

import java.util.ArrayList;

public class Client {

    private String nom;
    private int identifiant;
    private ArrayList <Reservation> reservations = new ArrayList
<Reservation>();
    private ReservationBuilder reservationBuilder;

    public ReservationBuilder getReservationBuilder() {
        return reservationBuilder;
    }

    public void setReservationBuilder(ReservationBuilder val) {
        this.reservationBuilder = val;
    }

    public int getIdentifiant() {
        return identifiant;
    }

    public void setIdentifiant(int val) {
        this.identifiant = val;
    }

    public String getNom() {
        return nom;
    }

    public void setNom(String val) {
        this.nom = val;
    }

    public void constructReservationVol(int numresa, voyage.Vol v,
Passager p) {
        reservationBuilder.createNewReservation(numresa);
        reservationBuilder.buildReservationVol(v, p);
        reservations.add(reservationBuilder.getReservation());
    }

    public void constructReservationTrajetRoutier(int numresa,
voyage.TrajetRoutier t, Passager p) {
        reservationBuilder.createNewReservation(numresa);
        reservationBuilder.buildReservationTrajetRoutier(t, p);
        reservations.add(reservationBuilder.getReservation());
    }

    public Reservation getReservation(int index) {
        return reservations.get(index);
    }
}

```

```

package voyage;

public abstract class AbstractVoyageFactory {

    public abstract Voyage create();
}

```

```

package main;

import reservation.*;
import voyage.*;

    public class Principal {

        static ReservationBuilder builder;

    public static void main(String[] args) {
        builder = ReservationBuilder.getInstance();

        Client client = new Client();
        client.setReservationBuilder(builder);
        client.setIdentifiant(10089);
        client.setNom("toto");

        VolFactory vf = VolFactory.getInstance();
        Vol v = vf.create();
        v.setDepart("Paris");
        v.setDestination("Tokyo");

        Passager p = new Passager("Untel");

        client.constructReservationVol(12, v, p);
        Reservation reservation = client.getReservation(0);
        reservation.affiche();
    }

}

```

```

package reservation;

import voyage.TrajetRoutier;
import voyage.Vol;

abstract class AbstractReservationBuilder {

    protected Reservation reservation;

    public Reservation getReservation() {
        return reservation;
    }

    public void createNewReservation(int val) {
        reservation = new Reservation();
        reservation.setIdentifiant(val);
    }

    public abstract void buildReservationVol(Vol v, Passager p);

    public abstract void
    buildReservationTrajetRoutier(TrajetRoutier t, Passager p);
}

```

```

package reservation;

public class Passager {

    private int identifiant;
    private String nom;

    public Passager() {
    }

    public Passager(String nom) {
        this.nom = nom;
    }

    public int getIdentifiant() {
        return identifiant;
    }

    private int testName() {
        if (this.nom == null)
            return 0 ;
        else
            return 1 ;
    }

    public void setIdentifiant(int val) {
        this.identifiant = val;
    }

    public String getNom() {
        if (testName() == 0)
            return "No Name" ;
        else
            return this.nom;
    }

    public void setNom(String val) {
        this.nom = val;
    }
}

```