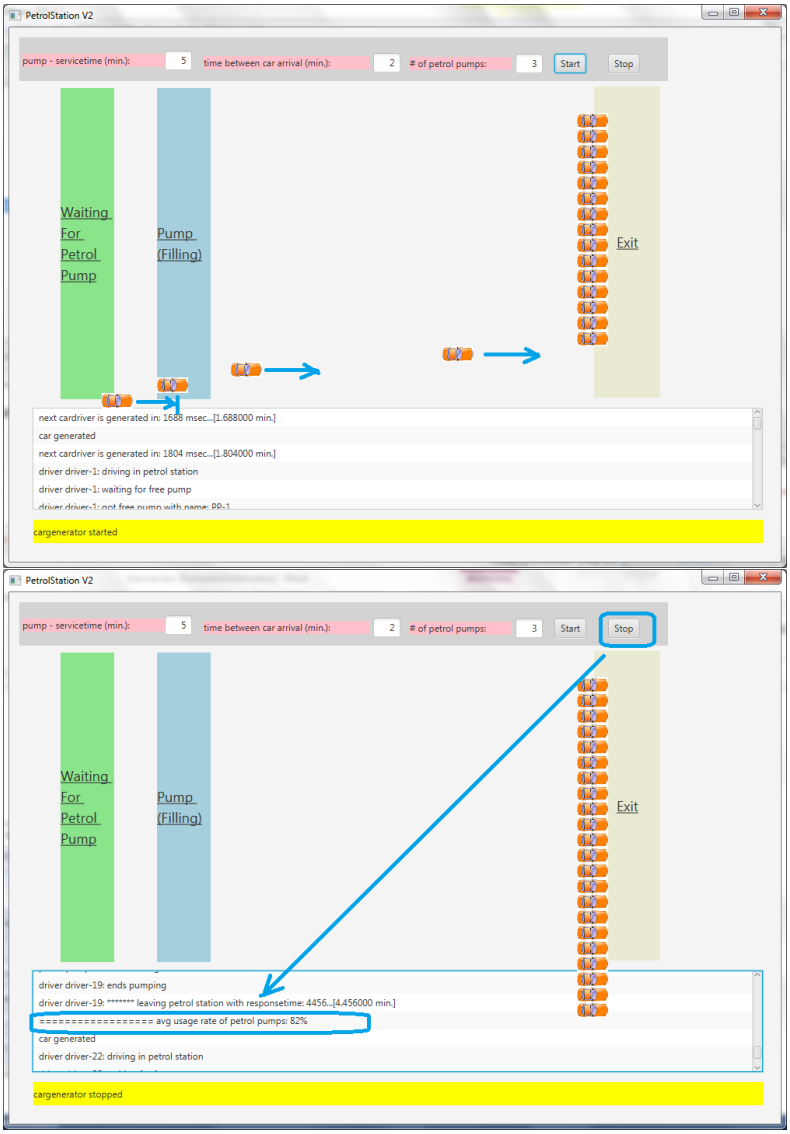
Overview

Simulation of a Petrol – Station

- animation of cars
- usage rate of pump stations

User's View



Hint for Developers

- **Control animation via workerprocess:**

Controller:

```
public void doAnimationMoving(CarDriver cd) throws Exception {
    ImageView iv = collImages.get(cd.getIdOfDriver()-1);
    Path path = new Path();
    //set start- and endposition
    path.getElements().add(new MoveTo(cd.getOldCooX(), START_Y + LANE_WIDTH * cd.getIdOfDriver()));
    path.getElements().add(new HLineTo(cd.getCurrentCooX()));
    PathTransition pathTransition = new PathTransition();
    pathTransition.setDuration(Duration.millis(ANIMATION_DURATION));
    pathTransition.setPath(path);
    pathTransition.setNode(iv);
    pathTransition.setAutoReverse(false);
    pathTransition.play();
}
```

Worker:

```
private void moveCar() {
    Platform.runLater(() -> {
        try {
            controller.doAnimationMoving(this);
        } catch (Exception ex) {
            Logger.getLogger(CarDriver.class.getName()).log(Level.SEVERE, null, ex);
        }
    });
}
```

- **Working with Constant Values**

If there are a lot of constants in use which are used in several classes,

- the constants should be defined in an interface
- and the interface is implemented by all classes, which need the constants.

```
public interface AnimCoordinates {

    public static final int START_Y = 100;
    public static final int HEIGHT = 20;
    public static final int WIDTH = 40;
    public static final int LANE_WIDTH = 20;
    public static final int XCOO_START = 10;
    public static final int XCOO_WAITING_PUMP = 60;
    public static final int XCOO_PUMPING = 210;
    public static final int XCOO_EXIT = 750;
}
```

➔

```
public class CarDriver extends Task<String> implements AnimCoordinates{
    private static Main_GUIController controller;
    private int idOfDriver = 0;
    private String nameOfDriver = null;
    private PetrolStation station = null;
    private Semaphore semaPetrolPumpFree = null;
    private final long timeStart;
    private long timeEnd;
    private ObservableList<String> obsList = null;
    private int currentCooX = XCOO_START;
    private int oldCooX = XCOO_START;

    public CarDriver(int idOfDriver, String name, PetrolStation station, Semaphore semaPetrolPumpFree) throws Exception {
        super();
        if (controller == null) {
            throw new Exception ("cardirver has no controller defined");
        }
```

Hint for Testers

Service-Time	Time-Between-Arrival	# of Pumps	Usage-rate per Pump (in theory:=ST/TBA/#)
5	5	2	50%
5	3	2	83%
5	2	3	83%