Main.java

*package* org.example;  
  
*import* javafx.application.Application;  
*import* javafx.fxml.FXMLLoader;  
*import* javafx.scene.Scene;  
*import* javafx.stage.Stage;  
*import* java.io.IOException;  
*import* java.net.URL;  
  
*public class* Main *extends* Application {  
 *@Override  
 public void* start(Stage stage) {  
 *try* {  
 FXMLLoader loader = *new* FXMLLoader(); *//Создаем загрузчик разметки* URL fxmlUrl = getClass().getResource("/App.fxml"); *//Создаем ссылку на разметку* loader.setLocation(fxmlUrl);  
 Scene root = loader.load(); *//Загружаем файл разметки окна* stage.setScene(root); *//Устанавливаем загруженную разметку на окно* stage.setResizable(*false*); *//Убираем возможноть менять размер окна* stage.setTitle("Cockroach runs"); *//Ставит заголовок окна* stage.show(); *//Показываем окно* } *catch* (IOException e) {  
 *throw new* RuntimeException(e);  
 }  
 }  
  
 *public static void* main(String[] args) {  
 *launch*();  
 }  
}

CarPool.java

*package* org.example.model;  
  
*import* javafx.application.Platform;  
*import* javafx.scene.canvas.Canvas;  
*import* javafx.scene.canvas.GraphicsContext;  
*import* javafx.scene.paint.Color;  
*import* java.util.ArrayList;  
*import* java.util.List;  
*import* java.util.function.Function;  
  
*public class* CarPool {  
 *private final* Canvas canvas;  
 *private final* List<Car> cars;  
 *private final* GraphicsContext context;  
  
 *private final int* START\_X = 200;  
 *private final int* FINISH\_X = 700;  
 *private final int* OBJ\_WIDTH = 100;  
  
 *private final* Function<Integer, ?> onWin;  
  
 *public* CarPool(Canvas canvas, Function<Integer, ?> onWin) {  
 *this*.canvas = canvas;  
 *this*.context = canvas.getGraphicsContext2D();  
 *this*.onWin = onWin;  
 *this*.cars = *new* ArrayList<>();  
 redraw();  
 }  
  
 *//Добавление потока в список  
 public void* addCar(Car car) {  
 car.setCarNumber(cars.size());  
 car.setPool(*this*);  
 cars.add(car);  
 }  
  
 *//Обновление позиции потоков на экране  
 public void* updateCarPosition(*int* num, *int* xPosition) {  
 *if* (cars.get(num).xPosition >= FINISH\_X - OBJ\_WIDTH) { *//Проверка на победу* onWin.apply(num); *//вызов функции для обработки победы* stop(); *//Остановка заезда  
 return*;  
 }  
  
 cars.get(num).xPosition += xPosition; *//Обновление позиции* Platform.*runLater*(() -> redraw()); *// Перерисовка* }  
  
 *//Метод перерисовки  
 private void* redraw() {  
 context.clearRect(0, 0, canvas.getWidth(), canvas.getHeight()); *//Очистка экрана* drawField();  
 *for* (*int* i = 0; i < cars.size(); ++i) { *//Отрисовка всех элементов в списке* context.setStroke(Color.*grayRgb*(i \* 50));  
 context.strokeOval(cars.get(i).xPosition, (i \* 110) + 20, OBJ\_WIDTH, OBJ\_WIDTH);  
 context.strokeText(String.*valueOf*(i), cars.get(i).xPosition + OBJ\_WIDTH / 2., (i \* 110) + 20 + OBJ\_WIDTH / 2.);  
 }  
 }  
  
 *//Отрисовка поля   
 private void* drawField() {  
 context.setStroke(Color.BLACK);  
 context.setLineWidth(1);  
 context.strokeText("Start", OBJ\_WIDTH, 10);  
 context.strokeText("Finish", FINISH\_X, 10);  
 context.setLineWidth(3);  
 context.strokeLine(OBJ\_WIDTH, 20, OBJ\_WIDTH, 580);  
 context.strokeLine(FINISH\_X, 20, FINISH\_X, 580);  
 }  
  
 *//Сброс заезда  
 public void* reset() {  
 stop();  
  
 *for* (Car car : cars)  
 car.xPosition = 0;  
  
 redraw();  
 }  
  
 *//Остановка заезда  
 public void* stop() {  
 *for* (Car car : cars)  
 car.pause();  
 redraw();  
 }  
  
 *//Начало заезда  
 public void* start() {  
 *for* (Car car : cars)  
 car.ride();  
 }  
}

Car.java

*package* org.example.model;  
  
*import* javafx.scene.image.Image;  
  
*import* java.util.concurrent.atomic.AtomicBoolean;  
  
*//Класс, представляющий поток  
public class* Car *extends* Thread {  
 *private int* carNumber;  
 *private* CarPool carPool;  
 *public final* Image image;  
 *public int* xPosition;  
  
 *private* AtomicBoolean running = *new* AtomicBoolean(*false*);  
  
 *public* Car(Image image) {  
 *this*.image = image;  
 xPosition = 0;  
 start();  
 }  
  
 *//Установка управляющего отрисовкой на экране класса  
 public void* setPool(CarPool carPool) {  
 *this*.carPool = carPool;  
 }  
  
 *//Установка номера машины для обновления в списке  
 public void* setCarNumber(*int* carNumber) {  
 *this*.carNumber = carNumber;  
 }  
  
 *//Остоновка потока  
 public void* pause() {  
 running.set(*false*);  
 }  
  
 *//Запуск потока  
 public void* ride() {  
 running.set(*true*);  
 }  
  
 *@Override  
 public void* run() {  
 *while* (*true*) {  
 *while* (running.get()) { *//Приостановка потока  
 try* {  
 *int* t = (*int*) (Math.*random*() \* 100); *//Генерируем время работы  
 long* startTime = System.*currentTimeMillis*(); *//Получаем время начала работы  
 while* (System.*currentTimeMillis*() - startTime < t) { *//Пока время текущей работы < сгенерированной* carPool.updateCarPosition(carNumber, 1); *//Обновляем свою позицию* Thread.*sleep*(1L); *//Засыпаем, чтобы не было слишком много обновлений позиции* }  
  
 Thread.*sleep*((*long*) (Math.*random*() \* 3000)); *//Засыпаем на случайное время* } *catch* (InterruptedException ignored) {  
 }  
 }  
 }  
 }  
}

AppController.java

*package* org.example.controllers;  
  
*import* javafx.application.Platform;  
*import* javafx.fxml.*FXML*;  
*import* javafx.scene.canvas.Canvas;  
*import* javafx.scene.control.Label;  
*import* javafx.scene.image.Image;  
*import* org.example.model.Car;  
*import* org.example.model.CarPool;  
  
*public class* AppController {  
  
 *@FXML* Canvas canvas;  
  
 *@FXML* Label winnerLabel;  
  
 CarPool carPool;  
  
 *@FXML  
 public void* initialize() {  
 carPool = *new* CarPool(canvas, integer -> { *//Создание объекта для управления потоками* Platform.*runLater*(() -> winnerLabel.setText("Winner is: " + integer));  
 *return* 0;  
 });  
 Image carImage = *new* Image("Car.png");  
  
 *//Создание потоков* carPool.addCar(*new* Car(carImage));  
 carPool.addCar(*new* Car(carImage));  
 carPool.addCar(*new* Car(carImage));  
 carPool.addCar(*new* Car(carImage));  
 carPool.addCar(*new* Car(carImage));  
  
 carPool.reset();  
 }  
  
  
 *//Обработчики нажатий на кнопки  
 @FXML  
 public void* start() {  
 carPool.start();  
 }  
  
 *@FXML  
 public void* reset() {  
 carPool.reset();  
 }  
  
 *@FXML  
 public void* about() {  
 AboutController aboutController = *new* AboutController(canvas.getScene().getWindow());  
 aboutController.show();  
 }  
  
 *@FXML  
 public void* exit() {  
 carPool.reset();  
 Platform.*exit*();  
 }  
}

App.fxml

<?xml version="1.0" encoding="UTF-8"?>  
  
<?import javafx.scene.control.ButtonType?>  
<?import javafx.scene.control.DialogPane?>  
<?import javafx.scene.control.Label?>  
<?import javafx.scene.image.Image?>  
<?import javafx.scene.image.ImageView?>  
<?import javafx.scene.layout.VBox?>  
  
<*DialogPane* onMouseClicked="#closeDialog" prefHeight="308.0" prefWidth="300.0" xmlns="http://javafx.com/javafx/19" xmlns:fx="http://javafx.com/fxml/1">  
 <*content*>  
 <*VBox* alignment="TOP\_CENTER" maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="264.0" prefWidth="300.0">  
 <*children*>  
 <*ImageView* fitHeight="150.0" fitWidth="200.0" pickOnBounds="true" preserveRatio="true">  
 <*image*>  
 <*Image* url="@cockroach.jpg" />  
 </*image*>  
 </*ImageView*>  
 <*Label* prefHeight="49.0" prefWidth="278.0" text="&quot;Cockroach runs&quot; can show you how threads works separately." textAlignment="CENTER" wrapText="true" />  
 <*Label* text=" This program is made by Maxim Varlamov." />  
 <*Label* text="© &quot;Cockroach runs&quot; 2023-2023" />  
 </*children*>  
 </*VBox*>  
 </*content*>  
 <*buttonTypes*>  
 <*ButtonType* fx:constant="CANCEL" />  
 </*buttonTypes*>  
</*DialogPane*>

AboutController.java

*package* org.example.controllers;  
  
*import* javafx.fxml.*FXML*;  
*import* javafx.fxml.FXMLLoader;  
*import* javafx.scene.control.Dialog;  
*import* javafx.scene.control.DialogPane;  
*import* javafx.stage.Modality;  
*import* javafx.stage.Window;  
  
*import* java.io.IOException;  
  
*public class* AboutController *extends* Dialog<Void> {  
 *public* AboutController(Window window) {  
 FXMLLoader loader = *new* FXMLLoader(); *//Загрузчик разметки* loader.setLocation(getClass().getResource("/About.fxml"));  
 loader.setController(*this*);  
  
 *try* {  
 DialogPane pane = loader.load();  
 initOwner(window); *//Устанавливаем родительский компонент* initModality(Modality.APPLICATION\_MODAL); *//Устанавливаем режим диалогового окна* setResizable(*true*);  
 setTitle("About Cockroach runs");  
 setDialogPane(pane);  
 getDialogPane()  
 .getScene()  
 .getWindow()  
 .setOnCloseRequest(event -> closeDialog()); *//Обрабатываем событие закрытия окна* } *catch* (IOException e) {  
 *throw new* RuntimeException(e);  
 }  
 }  
  
 *//Обработчик закрытия окна  
 @FXML  
 public void* closeDialog() {  
 getDialogPane().getScene().getWindow().hide();  
 }  
}

About.fxml

<?xml version="1.0" encoding="UTF-8"?>  
  
<?import javafx.scene.control.ButtonType?>  
<?import javafx.scene.control.DialogPane?>  
<?import javafx.scene.control.Label?>  
<?import javafx.scene.image.Image?>  
<?import javafx.scene.image.ImageView?>  
<?import javafx.scene.layout.VBox?>  
  
<*DialogPane* onMouseClicked="#closeDialog" prefHeight="308.0" prefWidth="300.0" xmlns="http://javafx.com/javafx/19" xmlns:fx="http://javafx.com/fxml/1">  
 <*content*>  
 <*VBox* alignment="TOP\_CENTER" maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="264.0" prefWidth="300.0">  
 <*children*>  
 <*ImageView* fitHeight="150.0" fitWidth="200.0" pickOnBounds="true" preserveRatio="true">  
 <*image*>  
 <*Image* url="@cockroach.jpg" />  
 </*image*>  
 </*ImageView*>  
 <*Label* prefHeight="49.0" prefWidth="278.0" text="&quot;Cockroach runs&quot; can show you how threads works separately." textAlignment="CENTER" wrapText="true" />  
 <*Label* text=" This program is made by Maxim Varlamov." />  
 <*Label* text="© &quot;Cockroach runs&quot; 2023-2023" />  
 </*children*>  
 </*VBox*>  
 </*content*>  
 <*buttonTypes*>  
 <*ButtonType* fx:constant="CANCEL" />  
 </*buttonTypes*>  
</*DialogPane*>