Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Новосибирский государственный технический университет»

Кафедра вычислительной техники

**Отчет**

По лабораторной работе №2

По дисциплине: «Технологии и методы программирования»

Основы программирования на Java.

Разработка графического интерфейса приложения.

Вариант 8

Выполнили: Павлюк А.С. Преподаватель: Копылова О.А.

Казанцев К.О.

Группа: АВТ-008

Факультет: АВТФ

Новосибирск, 2022

**Цель работы:**

Познакомиться с основными графическими библиотеками Java - AWT и Swing и их основными компонентами. Изучить классы менеджеров компоновки, классы создания меню приложения, основных и диалоговых окон и т.д.

**Задание к лабораторной работе:**

Доработать программу, созданную в лабораторной работе № 1:

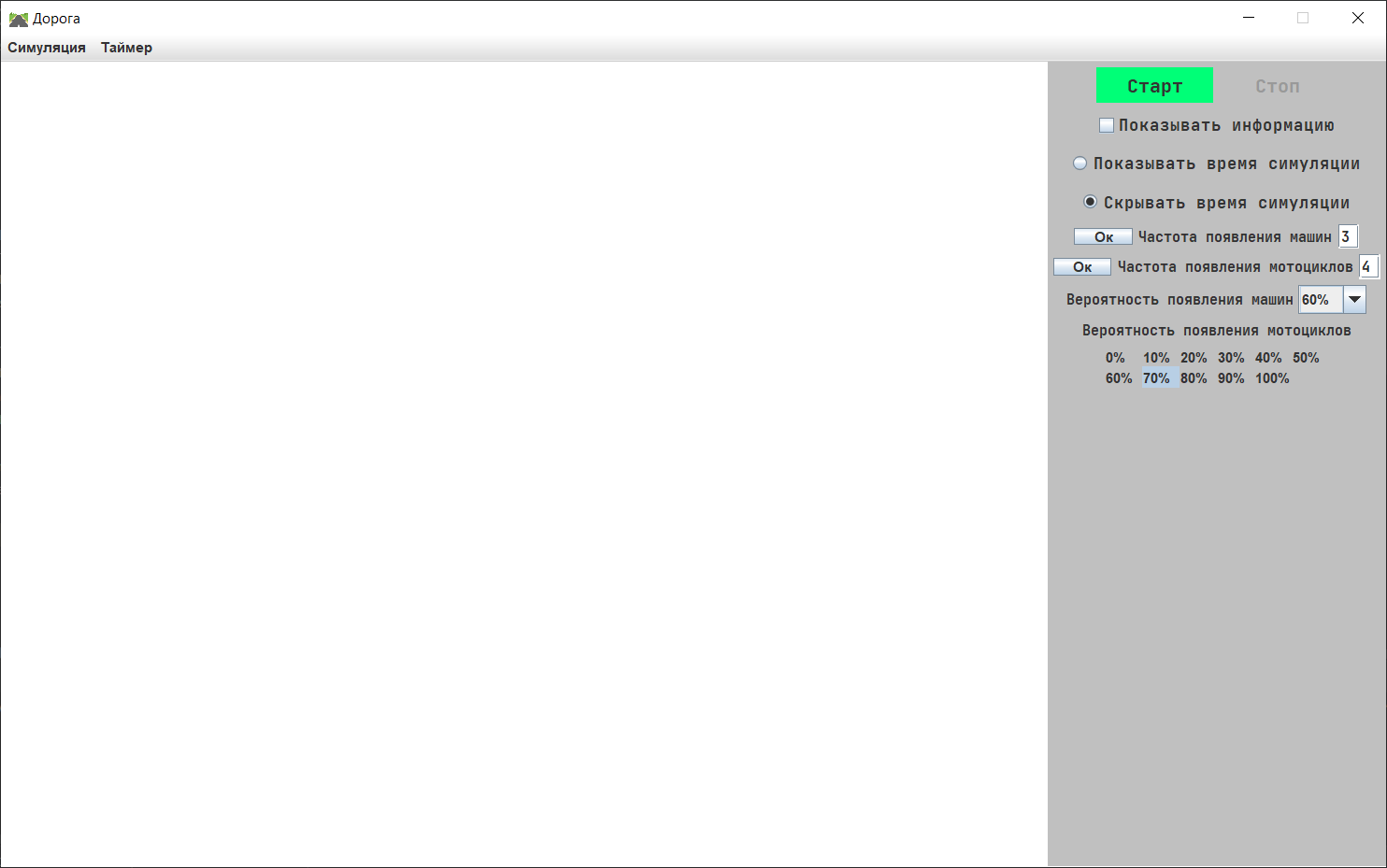
* 1. поделить рабочую область окна приложения на 2 части. Визуализация переносится в одну часть окна, панель управления в другую;
  2. добавить кнопки «Старт» и «Стоп» в панель управления. Они должны запускать и останавливать симуляцию соответственно. Если симуляция остановлена, то кнопка «Стоп» должна блокироваться. Если симуляция идет, то блокируется кнопка «Старт». Клавиши **B** и **E** должны функционировать по-прежнему;
  3. добавить переключатель «Показывать информацию», который разрешает отображение модального диалога из 7 пункта задания;
  4. добавить группу из 2 исключающих переключателей: «Показывать время симуляции» и «Скрывать время симуляции». Клавиша **T** должна функционировать по-прежнему;
  5. используя различные менеджеры компоновки, реализовать интерфейс пользователя согласно индивидуальному заданию;
  6. добавить в программу главное в меню и панель инструментов, в которых продублировать основные команды вашего интерфейса пользователя;
  7. при остановке симуляции должно появляться модальное диалоговое окно (при условии, что оно разрешено) с информацией о количестве и типе сгенерированных объектов, а также времени симуляции. Вся информация выводится в элементе TextArea, недоступном для редактирования. В диалоговом окне должно быть 2 кнопки: «ОК» и «Отмена». При нажатии на «ОК» симуляции останавливается, а при нажатии на «Отмена», соответственно продолжается;
  8. предусмотреть проверку данных вводимых пользователем. При вводе неверного значения обрабатывать исключительную ситуацию: выставлять значение по умолчанию и выводить диалоговое окно с сообщением об ошибке;
  9. Реализовать следующие элементы управления:
  + Периоды рождения объектов – текстовые поля;
  + Для задания вероятностей рождения объектов комбобокс и список (шаг значений 10%);
  + Дополнить интерфейс поясняющими метками.

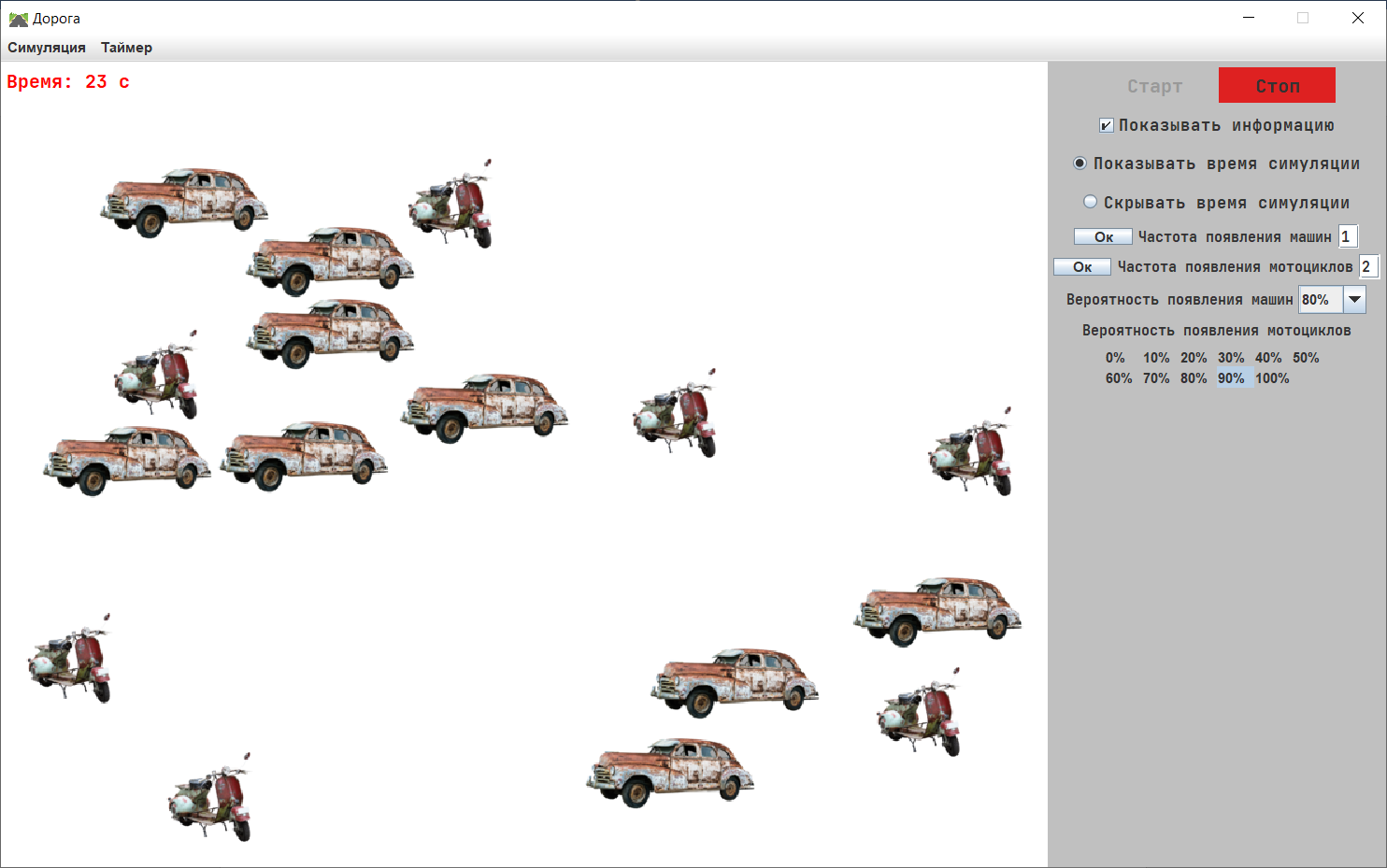
**Описание программы:**

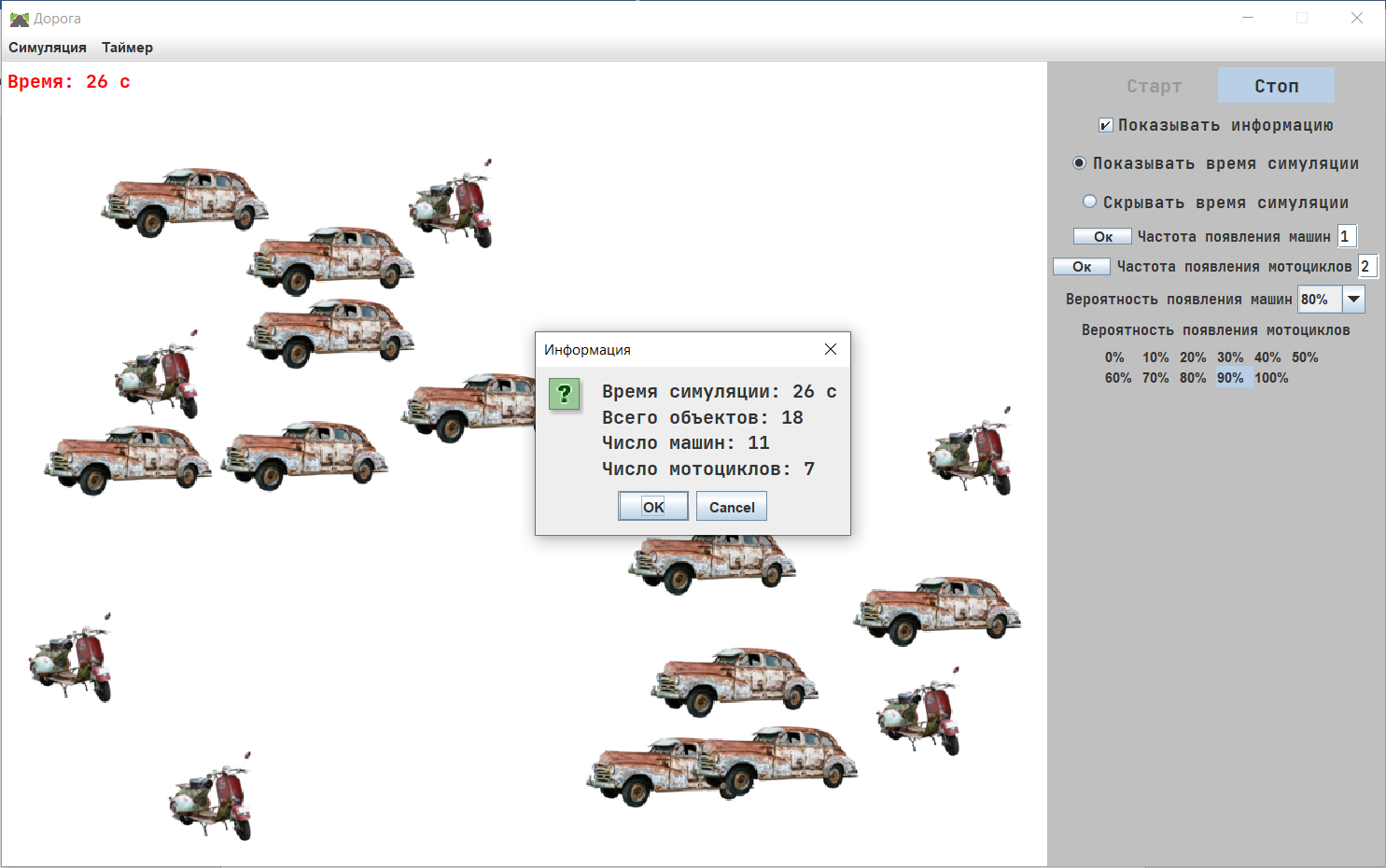
В класс MyFrame была добавлена реализация панели управления.

На панель управления добавлены кнопки «Старт» и «Стоп». Кнопка «Старт» запускает симуляцию, если она не запущена. Кнопка «Стоп» вызывает диалоговое окно, предоставляющее выбор – продолжить симуляцию по кнопке «Cancel», или прекратить симуляцию по кнопке «OK». Также данное диалоговое окно выводит информацию по симуляции, если прожат соответствующий флажок, находящийся ниже кнопок запуска и остановки симуляции. Ниже расположены взаимоисключающие переключатели, отвечающие за отображение таймера. Далее располагаются текстовые поля для задания частоты появления. Следом идут выпадающий и обычный списки, задающие вероятность появления объекта. Также было добавлено меню, в котором дублируются основные элементы управления.

**Пример выполнения работы**







**Вывод:** в ходе выполнения лабораторной работы мы познакомились с функционалом графической библиотеки Swing. В приложение была добавлена панель управления для более удобного взаимодействия пользователя с программой.

**Приложение**

**Изменения в классе MyFrame**

JButton **start**;  
 JButton **stop**;  
 JCheckBox **showInfo**;  
 JRadioButton **showTimer**;  
 JRadioButton **hideTimer**;  
 JOptionPane **dialog**;  
 JButton **submitCar**;  
 JButton **submitMoto**;  
 JTextField **carsFreqText**;  
 JTextField **motoFreqText**;  
 JComboBox<String> **carProbability**;  
 JList<String> **motoProbability**;  
  
 JMenuItem **startItem**;  
 JMenuItem **stopItem**;  
 JMenuItem **showInfoItem**;  
 JMenuItem **showTimerItem**;  
 JMenuItem **hideTimerItem**;  
  
 **public void** startSimulation() {  
 **if** (!**isStarted**) {  
 **start**.setEnabled(**false**);  
 **startItem**.setEnabled(**false**);  
 **stop**.setEnabled(**true**);  
 **stopItem**.setEnabled(**true**);  
 **start**.setContentAreaFilled(**false**);  
 **stop**.setContentAreaFilled(**true**);  
 System.***out***.println(**"---------------------------"**);  
 System.***out***.println(**"Car: chance - "** + **h**.**P1** + **"%, time - "** + **h**.**N1** +  
 **"\nBike: chance - "** + **h**.**P2** + **"%, time - "** + **h**.**N2**);  
 repaint();  
  
 **isStarted** = **true**;  
 **h**.**carCount** = 0;  
 **h**.**motoCount** = 0;  
 **timer** = **new** Timer();  
 **time** = 0;  
 **timer**.schedule(**new** TimerTask() {  
 @Override  
 **public void** run() {  
 **h**.update(**time**);  
 add(**new** JComponent() {  
 });  
 **time**++;  
 **timeLabel**.setText(**"Время: "** + **time** + **" с"**);  
 repaint();  
 }  
 }, 0, 1000);  
 }  
 }  
  
 **public void** stopSimulation() {  
 **timer**.cancel();  
 **dialog** = **new** JOptionPane();  
 JTextArea stats = **new** JTextArea(  
 **"Время симуляции: "** + **time** + **" c"** +  
 **"\nВсего объектов: "** + Habitat.***vehicles***.size() +  
 **"\nЧисло машин: "** + **h**.**carCount** +  
 **"\nЧисло мотоциклов: "** + **h**.**motoCount** );  
 stats.setBackground(**dialog**.getBackground());  
 stats.setEditable(**false**);  
 stats.setFont(**new** Font(**"JetBrains Mono"**, Font.***BOLD***, 16));  
  
 **int** n;  
 **if** (**showInfo**.isSelected() || **showInfoItem**.isSelected()) {  
 **showInfoItem**.setSelected(**true**);  
 n = JOptionPane.*showConfirmDialog*(**this**, stats, **"Информация"**, JOptionPane.***OK\_CANCEL\_OPTION***);  
 } **else** {  
 **showInfoItem**.setSelected(**false**);  
 n = JOptionPane.*showConfirmDialog*(**this**, **"Завершить симуляцию?"**, **"Завершение..."**, JOptionPane.***OK\_CANCEL\_OPTION***);  
 }  
  
 **if** (n == JOptionPane.***YES\_OPTION***) {  
 **showTimer**.setSelected(**false**);  
 **showTimerItem**.setSelected(**false**);  
 **hideTimer**.setSelected(**false**);  
 **hideTimerItem**.setSelected(**false**);  
 **if** (**isStarted**) {  
 **stop**.setEnabled(**true**);  
 **stopItem**.setEnabled(**true**);  
 **timer**.cancel();  
 **isStarted** = **false**;  
 }  
  
 **if** (**willShowTime**) {  
 **timeLabel**.setVisible(**false**);  
 **willShowTime** = **false**;  
 }  
  
 Habitat.***vehicles***.clear();  
 **start**.setEnabled(**true**);  
 **startItem**.setEnabled(**true**);  
 **stop**.setEnabled(**false**);  
 **stopItem**.setEnabled(**false**);  
 **start**.setContentAreaFilled(**true**);  
 **stop**.setContentAreaFilled(**false**);  
  
 } **else** {  
 **timer** = **new** Timer();  
 **timer**.schedule(**new** TimerTask() {  
 @Override  
 **public void** run() {  
 **h**.update(**time**);  
 add(**new** JComponent() {  
 });  
 **time**++;  
 **timeLabel**.setText(**"Время: "** + **time** + **" с"**);  
 repaint();  
 }  
 }, 0, 1000);  
 }  
  
 }  
  
 **public void** getTimer() {  
 **if** (!**willShowTime**) {  
 **timeLabel**.setVisible(**true**);  
 **willShowTime** = **true**;  
 **showTimer**.setSelected(**true**);  
 **showTimerItem**.setSelected(**true**);  
 **hideTimer**.setSelected(**false**);  
 **hideTimerItem**.setSelected(**false**);  
 } **else** {  
 **timeLabel**.setVisible(**false**);  
 **willShowTime** = **false**;  
 **hideTimer**.setSelected(**true**);  
 **hideTimerItem**.setSelected(**true**);  
 **showTimer**.setSelected(**false**);  
 **showTimerItem**.setSelected(**false**);  
 }  
 }  
  
 **start** = **new** JButton(**"Старт"**);  
 **start**.setBorderPainted(**false**);  
 **start**.setBackground(**new** Color(0, 255, 119));  
 **start**.setPreferredSize(**new** Dimension(100, 30));  
 **start**.setFont(**new** Font(**"JetBrains Mono"**, Font.***BOLD***, 16));  
 **start**.addActionListener(e -> startSimulation());  
 **start**.setFocusable(**false**);  
 **start**.setContentAreaFilled(**true**);  
 **panel**.add(**start**);  
  
 **stop** = **new** JButton(**"Стоп"**);  
 **stop**.setBorderPainted(**false**);  
 **stop**.setBackground(**new** Color(222, 33, 33));  
 **stop**.addActionListener(e -> stopSimulation());  
 **stop**.setEnabled(**false**);  
 **stop**.setPreferredSize(**new** Dimension(100, 30));  
 **stop**.setFont(**new** Font(**"JetBrains Mono"**, Font.***BOLD***, 16));  
 **stop**.setFocusable(**false**);  
 **panel**.add(**stop**);  
 **stop**.setContentAreaFilled(**false**);  
  
 **showInfo** = **new** JCheckBox(**"Показывать информацию"**);  
 **showInfo**.setFont(**new** Font(**"JetBrains Mono"**, Font.***BOLD***, 14));  
 **showInfo**.setBackground(**panel**.getBackground());  
 **showInfo**.addActionListener(e -> **showInfoItem**.setSelected(**showInfo**.isSelected()));  
 **showInfo**.setFocusable(**false**);  
 **panel**.add(**showInfo**);  
  
 **showTimer** = **new** JRadioButton(**"Показывать время симуляции"**);  
 **showTimer**.setFont(**new** Font(**"JetBrains Mono"**, Font.***BOLD***, 14));  
 **showTimer**.setBackground(**panel**.getBackground());  
 **showTimer**.addActionListener(e -> getTimer());  
 **showTimer**.setFocusable(**false**);  
 **panel**.add(**showTimer**);  
  
 **hideTimer** = **new** JRadioButton(**"Скрывать время симуляции"**);  
 **hideTimer**.setFont(**new** Font(**"JetBrains Mono"**, Font.***BOLD***, 14));  
 **hideTimer**.setBackground(**panel**.getBackground());  
 **hideTimer**.addActionListener(e -> getTimer());  
 **hideTimer**.setFocusable(**false**);  
 **hideTimer**.setSelected(**true**);  
 **panel**.add(**hideTimer**);  
  
 **submitCar** = **new** JButton(**"Ок"**);  
 **submitCar**.setPreferredSize(**new** Dimension(50, 15));  
 JLabel carsFreq = **new** JLabel(**"Частота появления машин"**);  
 **carsFreqText** = **new** JTextField(**""** + **h**.**N1**, 2);  
 carsFreq.setFont(**new** Font(**"JetBrains Mono"**, Font.***BOLD***, 12));  
 **carsFreqText**.setFont(**new** Font(**"JetBrains Mono"**, Font.***BOLD***, 12));  
 **submitCar**.addActionListener(e -> {  
 **try** {  
 **h**.**N1** = Integer.*parseInt*(**carsFreqText**.getText());  
 **if** (**h**.**N1** <= 0) **throw new** Exception();  
 } **catch** (Exception exp) {  
 JOptionPane.*showMessageDialog*(**this**, **"Введите целое положительое число!"**);  
 **h**.**N1** = 3;  
 System.***out***.println(**"Поймано исключение "** + exp.getMessage());  
 **carsFreqText**.setText(**""** + **h**.**N1**);  
 }  
 });  
 **submitCar**.setFocusable(**false**);  
 **panel**.add(**submitCar**);  
 **panel**.add(carsFreq);  
 **panel**.add(**carsFreqText**);  
  
 **submitMoto** = **new** JButton(**"Ок"**);  
 **submitMoto**.setPreferredSize(**new** Dimension(50, 15));  
 JLabel motoFreq = **new** JLabel(**"Частота появления мотоциклов"**);  
 **motoFreqText** = **new** JTextField(**""** + **h**.**N2**, 2);  
 motoFreq.setFont(**new** Font(**"JetBrains Mono"**, Font.***BOLD***, 12));  
 **motoFreqText**.setFont(**new** Font(**"JetBrains Mono"**, Font.***BOLD***, 12));  
 **submitMoto**.addActionListener(e -> {  
 **try** {  
 **h**.**N2** = Integer.*parseInt*(**motoFreqText**.getText());  
 **if** (**h**.**N2** <= 0) **throw new** Exception();  
 } **catch** (Exception exp) {  
 JOptionPane.*showMessageDialog*(**this**, **"Введите целое положительое число!"**);  
 **h**.**N2** = 4;  
 System.***out***.println(**"Поймано исключение "** + exp.getMessage());  
 **motoFreqText**.setText(**""** + **h**.**N2**);  
 }  
 });  
 **submitMoto**.setFocusable(**false**);  
 **panel**.add(**submitMoto**);  
 **panel**.add(motoFreq);  
 **panel**.add(**motoFreqText**);  
  
 String[] prob = {  
 **"0%"**,  
 **"10%"**,  
 **"20%"**,  
 **"30%"**,  
 **"40%"**,  
 **"50%"**,  
 **"60%"**,  
 **"70%"**,  
 **"80%"**,  
 **"90%"**,  
 **"100%"**,  
 };  
  
 JLabel carsP = **new** JLabel(**"Вероятность появления машин"**);  
 carsP.setFont(**new** Font(**"JetBrains Mono"**, Font.***BOLD***, 12));  
 **panel**.add(carsP);  
 **carProbability** = **new** JComboBox<>(prob);  
 **carProbability**.setSelectedItem(**""** + **h**.**P1** + **"%"**);  
 **panel**.add(**carProbability**);  
  
 **carProbability**.addActionListener(e -> {  
 **if** (e.getSource() == **carProbability**) {  
 String prb = Objects.*requireNonNull*(**carProbability**.getSelectedItem()).toString().replace(**"%"**, **""**);  
 **h**.**P1** = Integer.*parseInt*(prb);  
 }  
 });  
 **carProbability**.setFocusable(**false**);  
  
 JLabel motoP = **new** JLabel(**"Вероятность появления мотоциклов"**);  
 motoP.setFont(**new** Font(**"JetBrains Mono"**, Font.***BOLD***, 12));  
 **panel**.add(motoP);  
 **motoProbability** = **new** JList<>(prob);  
 **motoProbability**.setLayoutOrientation(JList.***HORIZONTAL\_WRAP***);  
 **motoProbability**.setBackground(**panel**.getBackground());  
 **motoProbability**.setVisibleRowCount(2);  
 **motoProbability**.setSelectedIndex(7);  
 **motoProbability**.setSelectionMode(ListSelectionModel.***SINGLE\_SELECTION***);  
 **motoProbability**.addListSelectionListener(e -> {  
 **if** (e.getSource() == **motoProbability**) {  
 String prb = **motoProbability**.getSelectedValue().replace(**"%"**, **""**);  
 **h**.**P2** = Integer.*parseInt*(prb);  
 }  
 });  
 **motoProbability**.setFocusable(**false**);  
 **panel**.add(**motoProbability**);  
  
 JMenuBar menu = **new** JMenuBar();  
  
 JMenu simulationMenu = **new** JMenu(**"Симуляция"**);  
 **startItem** = **new** JMenuItem(**"Старт"**);  
 **stopItem** = **new** JMenuItem(**"Стоп"**);  
 **showInfoItem** = **new** JCheckBoxMenuItem(**"Показывать информацию"**);  
 **startItem**.addActionListener(e -> startSimulation());  
 **stopItem**.addActionListener(e -> stopSimulation());  
 **stopItem**.setEnabled(**false**);  
 **showInfoItem**.addActionListener(e -> **showInfo**.setSelected(**showInfoItem**.isSelected()));  
 simulationMenu.add(**startItem**);  
 simulationMenu.add(**stopItem**);  
 simulationMenu.add(**showInfoItem**);  
  
 JMenu timerMenu = **new** JMenu(**"Таймер"**);  
 **showTimerItem** = **new** JRadioButtonMenuItem(**"Показывать"**);  
 **hideTimerItem** = **new** JRadioButtonMenuItem(**"Скрывать"**);  
 **showTimerItem**.addActionListener(e -> getTimer());  
 **hideTimerItem**.addActionListener(e -> getTimer());  
 **hideTimerItem**.setSelected(**true**);  
 timerMenu.add(**showTimerItem**);  
 timerMenu.add(**hideTimerItem**);  
  
 menu.add(simulationMenu);  
 menu.add(timerMenu);  
  
 setJMenuBar(menu);  
 menu.setFocusable(**false**);  
  
 setVisible(**true**);  
 }  
}