**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»**

**ІНСТИТУТ ПІСЛЯДИПЛОМНОЇ ОСВІТИ (II В.О.)**



**КУРСОВА РОБОТА**

*з предмету:*

*«Об‘єктно-орієнтоване програмування»*

**Виконав:**

студент групи ПЗС-11

Гринчук Т. А.

**Прийняла:**

доц.Коротєєва Т.О.

ЛЬВІВ – 2014

**ЗМІСТ**

ВСТУП……………………………………………………………………………..3

1. ТЕОРЕТИЧНА ЧАСТИНА............................................................................4

1.1. Проектування структури бібліотек програмного продукту......................4

1.2. Проектування основних класів ....................................................................6

1.3. Взаємозв’язки між класами: діаграма класів ..............................................9

2. РОЗРОБКА ПРОГРАМИ..............................................................................10

2.1. Розробка користувацького інтерфейсу……………………….………....10

2.2. Алгоритми опрацювання даних………………………………..................13

2.3. Робота з файлами...........................................................................................16

3. ПРОТОКОЛ РОБОТИ ТА ІНСТРУКЦІЯ КОРИСТУВАЧА....................17

ВИСНОВКИ.............................................................................................................22

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.........................................................23

ДОДАТОК А. Лістинг програми............................................................................24

**ВСТУП**

Завданням даної курсової роботи є засобами ООП, на обраній мові програмування розробити програму, згідно варіанту технічного завдання та оформити відповідний звіт.

**Варіант 5.**

Створити таблицю:

№ | Марка машини | Колір | Ціна | Потужність

1. Відомим алгоритмом відсортувати записи за потужністю.
2. За заданою маркою авто визначити найдешевшу та найменш потужну
3. Вивести марки авто, в яких однакова ціна та різні кольори
4. Визначити марки авто, ціна яких не входить в задані межі.
5. В кожній марці авто визначити найпотужнішу червоного кольору та найдешевшу чорного кольору.

Для реалізації завдання я обрав мову програмування Java та середовище виконання NetBeans IDE 8.0.

Java — об'єктно-орієнтована мова програмування, випущена компанією Sun Microsystems у 1995 році як основний компонент платформи Java. Зараз мовою займається компанія Oracle, яка придбала Sun Microsystems у 2009 році. Синтаксис мови багато в чому схожий на C та C++. У офіційній реалізації, Java програми компілюються у байткод, який при виконанні інтерпретується віртуальною машиною для конкретної платформи.

Oracle надає компілятор Java та віртуальну машину Java, які задовольняють специфікації Java Community Process, під ліцензією GNU General Public License.

Мова значно запозичила синтаксис із C і C++. Зокрема, взято за основу об'єктну модель С++, проте її модифіковано. Усунуто можливість появи деяких конфліктних ситуацій, що могли виникнути через помилки програміста та полегшено сам процес розробки об'єктно-орієнтованих програм. Ряд дій, які в С/C++ повинні здійснювати програмісти, доручено віртуальній машині. Передусім, Java розроблялась як платформо-незалежна мова, тому вона має менше низькорівневих можливостей для роботи з апаратним забезпеченням. За необхідності таких дій java дозволяє викликати підпрограми, написані іншими мовами програмування.

NetBeans IDE - вільна інтегрована середовище розробки додатків (IDE) на мовах програмування Java , Python , PHP , JavaScript , C , C ++ , Ада та ряду інших.

Проект NetBeans IDE підтримується і спонсорується компанією Oracle, проте розробка NetBeans ведеться незалежним спільнотою розробників - ентузіастів (NetBeans Community ) і компанією NetBeans Org .

За якістю і можливостям останні версії NetBeans IDE не поступаються кращим комерційним (платним) інтегрованим середовищам розробки для мови Java, таким, як IntelliJ IDEA, підтримуючи рефакторінг, профілювання, виділення синтаксичних конструкцій кольором, автодоповнення набираються конструкцій на льоту, безліч визначених шаблонів коду та ін.

Для розробки програм в середовищі NetBeans і для успішної інсталяції та роботи самого середовища NetBeans повинен бути попередньо встановлений Sun JDK або J2EE SDK підходящої версії. Середовище розробки NetBeans за замовчуванням підтримує розробку для платформ J2SE і J2EE . Починаючи з версії 6.0 Netbeans підтримує розробку для мобільних платформ J2ME, C ++ і PHP без установки додаткових компонентів.

NetBeans Platform - платформа для розробки модульних настільних Swing додатків. NetBeans IDE містить все, що потрібно для розробки плагінів і додатків на основі NetBeans Platform. Додатки можуть динамічно завантажувати інші модулі. Будь-який додаток може включити модуль оновлення, щоб дозволити користувачам завантажувати оновлення для програм і модулів в роботі додатку .

NetBeans Platform пропонує багаторазово використовувані сервіси і модулі для настільних додатків, дозволяючи розробникам сфокусуватися на логіці програми.

1. **ТЕОРЕТИЧНА ЧАСТИНА**
   1. **Проектування структури бібліотек програмного продукту**

Для того щоб умовно розділити архітектуру програмного проекту та структурні одиниці, як бібліотеки (в Java – вони називаються пакетами), а потім на класи, потрібно керуватися певними принципами моделювання. В даному проекті я використав шаблон проектування та розробки програмного забеспечення – **MVC**.

**Модель-вид-контролер** (або **Модель-вигляд-контролер**, англ. **Model-view-controller, MVC**) — архітектурний шаблон, який використовується під час проектування та розробки програмного забезпечення.

Цей шаблон поділяє систему на три частини: модель даних, вигляд даних та керування. Застосовується для відокремлення даних (модель) від інтерфейсу користувача (вигляду) так, щоб зміни інтерфейсу користувача мінімально впливали на роботу з даними, а зміни в моделі даних могли здійснюватися без змін інтерфейсу користувача.

Мета шаблону — гнучкий дизайн програмного забезпечення, який повинен полегшувати подальші зміни чи розширення програм, а також надавати можливість повторного використання окремих компонентів програми. Крім того використання цього шаблону у великих системах призводить до певної впорядкованості їх структури і робить їх зрозумілішими завдяки зменшенню складності.

Архітектурний шаблон Модель-Вид-Контролер (MVC) поділяє програму на три частини. У тріаді до обов'язків компоненту Модель (Model) входить зберігання даних і забезпечення інтерфейсу до них. Вигляд (View) відповідальний за представлення цих даних користувачеві. Контролер (Controller) керує компонентами, отримує сигнали у вигляді реакції на дії користувача, і повідомляє про зміни компоненту Модель. Така внутрішня структура в цілому поділяє систему на самостійні частини і розподіляє відповідальність між різними компонентами.

MVC поділяє цю частину системи на три самостійні частини: введення даних, компонент обробки даних і виведення інформації. Модель, як вже було відмічено, інкапсулює ядро даних і основний функціонал з їх обробки. Також компонент Модель не залежить від процесу введення або виведення даних. Компонент виводу Вигляд може мати декілька взаємопов'язаних областей, наприклад, різні таблиці і поля форм, в яких відображається інформація. У функції Контролера входить моніторинг за подіями, що виникають в результаті дій користувача (зміна положення курсора миші, натиснення кнопки або введення даних в текстове поле).

Зареєстровані події транслюються в різні запити, що спрямовуються компонентам Моделі або об'єктам, відповідальним за відображення даних. Відокремлення моделі від вигляду даних дозволяє незалежно використовувати різні компоненти для відображення інформації. Таким чином, якщо користувач через Контролер внесе зміни до Моделі даних, то інформація, подана одним або декількома візуальними компонентами, буде автоматично відкоригована відповідно до змін, що відбулися

Концепція MVC вперше застосувалася при проектуванні мови програмування Smalltalk як модель для інтерфейсу користувача. Також в область застосування концепції входить реалізація каркаса Документ-Вид (Document-View) в рамках бібліотеки MFC для мови Visual C++ . У сучасних технологіях концепція MVC представлена схемою JSP Model 1/2 для динамічної обробки Web-змісту на основі Java Server Pages (JSP).

У мові програмування **Java** **концепція** **MVC** підтримується на рівні стандартних класів-бібліотек. В результаті використання парадигми MVC програміст отримує в своє розпорядження могутню структуру об'єктів-компонентів, функції яких чітко розмежовані, що гарантує надійність і розширюваність системи, що розробляється.

Веб-каркаси: *Struts, WebWork/Struts 2, Spring MVC, JavaServer Faces (JSF), Tapestry*

Десктоп-каркаси: *Swing, JFace*.

Враховуючи вищезгадане, архітектура проекту буде складатися з наступних бібліотек (пакетів):

* ***com.taras\_grynchuk.oop*** – містить класи, що містять в собі модель даних проекту;
* ***com.taras\_grynchuk.oop.forms*** – інтерфейс користувача(GUI – graphic user interface): візуальні форми та засоби відображення даних у зручному для користувача вигляді;
* ***com.taras\_grynchuk.oop.files*** – класи роботи з файлами.
  1. **Проектування основних класів**

Пакет ***com.taras\_grynchuk.oop***, містить наступні класи:

**Goods** – клас товарів. Є базовим класом в архітектурі структур даних проекту. Він містить атрибути та методи, які властиві для всіх товарів.

public abstract class Goods implements Serializable {

//атрибути товару

protected int uid; //унікальний ідентифікатор

protected String name; //Назва товару

protected float price; //ціна

//методи встановлення/читання атрибутів

public String getName() {...}

public void setName(String name) {...}

public float getPrice() {...}

public void setPrice(float price) {...}

//отримання артикулу

abstract int getArticle();

}

**Car** - клас автомобілів. Наслідується від класу **Goods** і доповнює його функціями та методами, які властиві лише автомобілям, як одиницям товару.

public class Car extends Goods implements Const, Serializable {

private static int count = 0; //загальна к-сть автомобілів

//атрибути автомобіля

private String color; //колір

private int power; //потужність

//------------------------------------------------

//методи встановлення/читання атрибутів

//загальна к-сть автомобілів

public static int getCount() {...}

public static void setCount(int newCount) {...}

//артикул (не можна змінювати)

@Override

public int getArticle() {...}

//марка

public String getModel() {...}

public void setModel(String model) {...}

//колір

public String getColor() {...}

public void setColor(String color) {...}

//потужність

public int getPower() {...}

public void setPower(int power) {...}

//конструктори

public Car(String model, String color, float price, int power) {...}

//пуста позицiя

public Car() {...}

//представлення об'єкта строкою

@Override

public String toString() {...}

}

**ListOfCars** – список автомобілів. Крім того, що є контейнером для всіх екземплярів класу Car, містить методи для розрахунку та обробки даних, такі як сотування, вибірка за конкретними властивостями, тощо.

public class ListOfCars extends LinkedList<Car> implements Serializable {

//сортування за потужністю "методом вибору"

public void sort() {...}

//список моделей без повторень

public String[] getArrayOfModels() {...}

//список кольорів без повторень

public String[] getArrayOfColors(String firstColor) {...}

public String[] getArrayOfColors() {...}

//найдешевша за вказаною моделлю

public int getMinPriceIndex(String model) {...}

//найдешевша за вказаною моделлю та кольором

public int getMinPriceIndex(String model, String color) {...}

//найменш потужна

public int getMinPowerIndex(String model) {...}

//марки з одинаковою ціною та різними кольорами

public String getSamePrice() {...}

//Марки, ціна яких не входить в задані межі

public String getOutOfBoundsPrices(float priceFrom, float priceTo) {...}

//найпотужніша за вказаним кольором та моделлю

public int getMaxPowerIndex(String model, String color) {...}

//найпотужніша та найдешевша за вказаними кольорами у кожній марці

public String getMaxPowerAndMinPrice(String maxPowerColor, String minPriceColor) {...}

}

**Інтерфейс** **Const** – константи проекту

public interface Const {

int FIRST\_CAR\_ARTICLE = 10000; //початковий артикул автомобілів

String EMPTY\_MODEL = "Chevrolet"; //марка по-замовчуванню

String EMPTY\_COLOR = "<не вибрано>"; //колір по замовчуванню

String BLACK\_COLOR = "чорний";

String RED\_COLOR = "червоний";

}

Пакет ***com.taras\_grynchuk.oop.forms***, містить наступні класи: **MainWindow, AboutBox, CheapestForm, MinMaxForm, PriceOutOfBoundsForm, SamePriceForm**. Ці класи є вікнами графічного інтерфейсу, кожне з, яких виконує певні функції.

Пакет ***com.taras\_grynchuk.oop.files***, містить наступні класи:

**SerializableFiles** – призначений для зберігання та відновлення класів структур даних на жорсткому диску у вигляді файлу. Він використовує механізм серіалізації класів.

public class SerializableFiles {

//відкрити файл з списком машин

public static ListOfCars openFromSerFile() throws IOException, ClassNotFoundException {...}

//зберегти список машин у файл

public static void saveToSerFile(ListOfCars list) throws IOException, ClassNotFoundException {...}

}

**TextFile** – призначений для імпорту/експорту списку машин у текстовий файл.

public class TextFile {

public static void exportToTextFile(ListOfCars list, String fileName) throws IOException {...}

public static void importFromTextFile(ListOfCars list, String fileName) throws IOException, NumberFormatException, Exception {...}

}

* 1. **Взаємозв’язки між класами: діаграма класів**



Рис. 1.1. UML-діаграма класів

Як видно з діаграми, виділяємо клас **Goods**, як батьківський, який має атрибути та методи, що наслідуються для дочірнього класу: **Car**. Клас **Car**, в свою чергу входить до контейнера **ListOfCars**, в якому зосередженні основні програмні засоби для виконання завдань проекту. Класи візуальних форм містять ссилку на **ListOfCars**, таким чином можна змінювати дані списка з цих форм. **TextFile**, **SerializableFiles** – «обслуговують» **ListOfCars** в частині збереження та відтворення цього класу з диска.

1. **РОЗРОБКА ПРОГРАМИ**

**2.1. Розробка користувацького інтерфейсу**

Для побудови віконного інтерфейсу в Java потрібно створити клас, який наслідує від класу JFrame бібліотеки javax.swing.

public class AboutBox extends javax.swing.JFrame {

public static void main(String args[]) {

}

}

Додамо до цієї форми інші візуальні компоненти, як кнопки та написи:

private javax.swing.JButton jButton1;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

Для того щоб кнопка почала реагувати на натиск, потрібно їй призначити «слухача»:

jButton1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton1ActionPerformed(evt);

}

});

………………….

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

dispose();

}

Також слід визначити початкові параметри для всіх компонентів та їх вирівнювання на формі. Після цих дій код класу набере наступного вигляду:

public class AboutBox extends javax.swing.JFrame {

public AboutBox() {

initComponents();

}

private void initComponents() {

jLabel1 = new javax.swing.JLabel();

jLabel3 = new javax.swing.JLabel();

jLabel4 = new javax.swing.JLabel();

jButton1 = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE\_ON\_CLOSE);

setTitle("Про програму");

setAlwaysOnTop(true);

setResizable(false);

jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/com/taras\_grynchuk/oop/forms/Portfolio.png"))); // NOI18N

jLabel3.setFont(new java.awt.Font("Tahoma", 2, 18)); // NOI18N

jLabel3.setForeground(new java.awt.Color(0, 153, 153));

jLabel3.setText("(с) 2014, Тарас Гринчук");

jLabel4.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N

jLabel4.setForeground(new java.awt.Color(0, 153, 153));

jLabel4.setText("Курсова робота з ООП");

jButton1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/com/taras\_grynchuk/oop/forms/Info.png"))); // NOI18N

jButton1.setText("ОК");

jButton1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton1ActionPerformed(evt);

}

});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(25, 25, 25)

.addComponent(jLabel1)

.addGap(33, 33, 33)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED\_SIZE, 218, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel4, javax.swing.GroupLayout.PREFERRED\_SIZE, 218, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addContainerGap(60, Short.MAX\_VALUE))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jButton1, javax.swing.GroupLayout.PREFERRED\_SIZE, 115, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(140, 140, 140))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(51, 51, 51)

.addComponent(jLabel1))

.addGroup(layout.createSequentialGroup()

.addGap(40, 40, 40)

.addComponent(jLabel4, javax.swing.GroupLayout.PREFERRED\_SIZE, 27, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(29, 29, 29)

.addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED\_SIZE, 27, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 44, Short.MAX\_VALUE)

.addComponent(jButton1, javax.swing.GroupLayout.PREFERRED\_SIZE, 43, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap())

);

pack();

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

dispose();

}

public static void main(String args[]) {

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new AboutBox().setVisible(true);

}

});

}

private javax.swing.JButton jButton1;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

}

Запустимо даний клас на виконання:

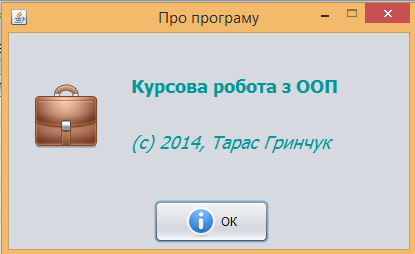


Рис. 2.1. Клас AboutBox у виконанні

**2.2. Алгоритми опрацювання даних**

Для пошуку максимальних та мінімальних цін та потужностей, скористаємось пря мим обходом списку машин:

public int getMinPriceIndex(String model, String color) {

float min = Float.MAX\_VALUE;

int index = -1;

for(int i = 0; i < size(); i++) {

Car c = get(i);

if(!model.toLowerCase().equals(c.getModel().toLowerCase())) continue;

if(color.length() > 0)

if(!color.toLowerCase().equals(c.getColor().toLowerCase())) continue;

float x = c.getPrice();

if(x < min) {

min = x;

index = i;

}

}

return index;

}

Для сортування використаємо алгоритм простої вибірки:

public void sort() {

for(int i = 0; i < size(); i++) {

// Знайдемо мінімальний елемент на

// проміжку індексів [i; size);

// спочатку його індекс рівний i

int minValueIndex = i;

// Перебираємо елементи, що залишилися на проміжку

for(int j = i + 1; j < size(); j++)

// Якщо елемент в позиції j менший

// елемента в позиції minValueIndex, то

// необхідно обновити значення індекса

if(get(j).getPower()

< get(minValueIndex).getPower()) minValueIndex = j;

// Міняємо поточний елемент з мінімальним

Collections.swap(this, i, minValueIndex);

}

}

Для вибірки едементів без повторів будемо використовувати таку структуру даних, як множина (HashSet):

//марки з одинаковою ціною та різними кольорами

public String getSamePrice() {

StringBuilder s = new StringBuilder();

String[] models = getArrayOfModels();

//йдемо по моделях

for(String model: models) {

//створимо множину цін даної марки

HashSet<Float> prices = new HashSet<>();

for(Car car: this)

if(car.getModel().equals(model)) prices.add(car.getPrice());

String strPrices = "";

//йдемо по цінах

for(float price: prices) {

//створимо множину кольорів даної марки та даної ціни

HashSet<String> colors = new HashSet<>();

for(Car car: this)

if(car.getModel().equals(model)) //перевірка на модель

if(car.getPrice() == price) //перевірка на ціну

colors.add(car.getColor());

String strColors = "";

//якщо кольори різні (тобто їх к-сть в множині > 1)

if(colors.size() > 1) {

for(String color: colors) {

if(!strColors.equals("")) strColors += ", ";

strColors += color;

}

if(strPrices.length() > 0) strPrices += "\n";

strPrices += "\tЦіна: " + price + " - ( " + strColors + " )";

}

}

if(strPrices.length() > 0) {

if(s.length() != 0) s.append("\n");

s.append("==============================================");

s.append("\nМарка: " + model);

s.append("\n" + strPrices);

}

}

return s.toString();

}

**2.3. Робота з файлами**

Для читання з текстових файлів в Java використовуються класи BufferedReader та FileReader:

public static void importFromTextFile(ListOfCars list, String fileName) throws IOException, NumberFormatException, Exception {

BufferedReader in = new BufferedReader(new FileReader(fileName));

//new File(fileName).getAbsoluteFile()));

String s;

while((s = in.readLine()) != null) {

String[] a = s.split("\t");

Car car = new Car(a[0], a[1], Float.parseFloat(a[2]), Integer.parseInt(a[3]));

list.add(car);

}

in.close()

}

Для запису в текстові файли використаємо клас PrintWriter:

public static void exportToTextFile(ListOfCars list, String fileName) throws IOException {

PrintWriter out = new PrintWriter(

new File(fileName).getAbsoluteFile());

for(Car car: list) {

char tab = (char) 9;

String s = "" +

car.getModel() + tab +

car.getColor() + tab +

car.getPrice() + tab +

car.getPower();

out.println(s);

}

out.close();

}

1. **ПРОТОКОЛ РОБОТИ ТА ІНСТРУКЦІЯ КОРИСТУВАЧА**

Основне вікно програми має вигляд (рис. 3.1):

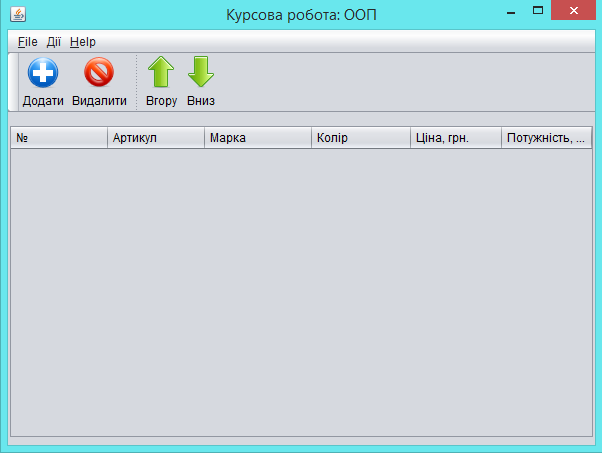


Рис. 3.1. Основне вікно програми

Для введення даних вручну, потрібно використовувати кнопки основної панелі: **Додати** та **Видалити**. Після натиску на **Додати** зявиться запис з початковими даними (рис. 3.2.).

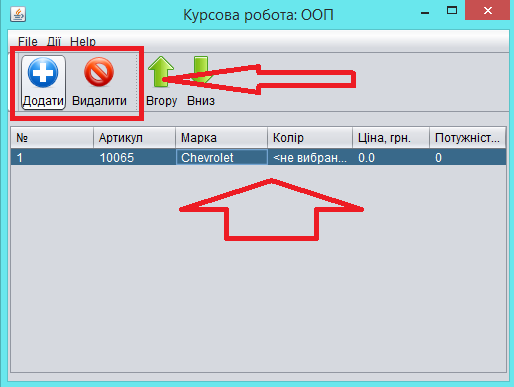


Рис. 3.2. Введення даних з клавіатури

Для імпорту даних з текстового файлу, потрібно скористатись меню Файл/Імпорт (рис. 3.3) та вибрати у діалогу (рис. 3.4) потрібний файл.

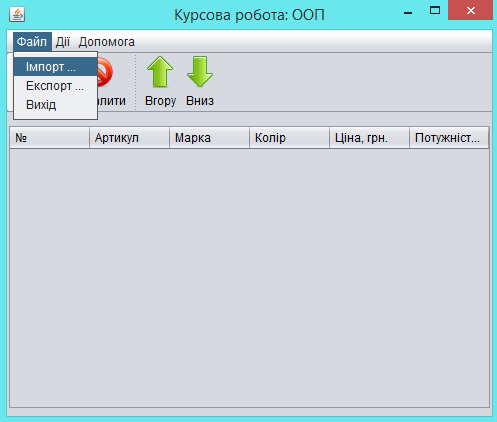


Рис. 3.3. Імпорт з файлу

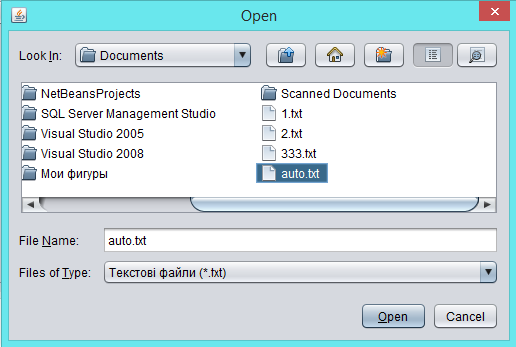


Рис. 3.4. Вибір файлу в діалозі

Таблиця наповниться даними з файлу (рис. 3.5):

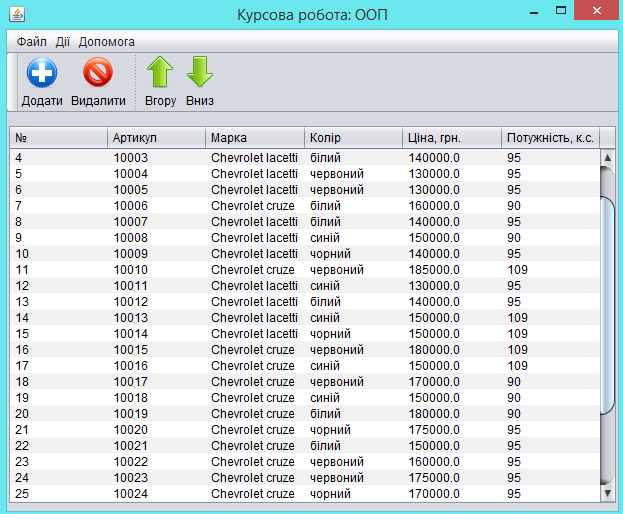


Рис. 3.5. Головне вікно після імпорту

Для експорту у файл скористаємось пунктом меню Файл/Експорт. Такий файл матиме вигляд (рис. 3.6):

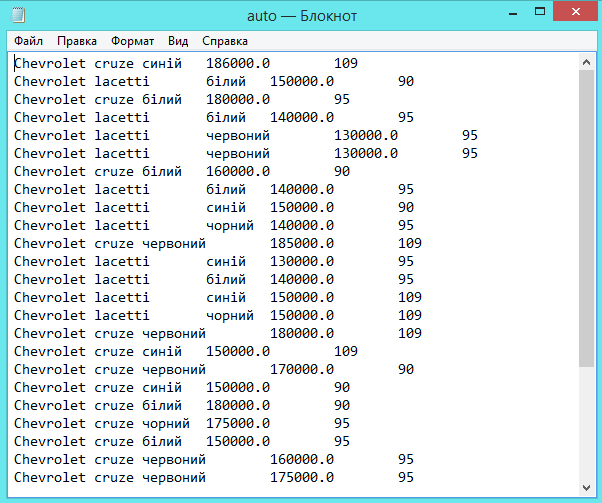


Рис. 3.6. Експортований текстовий файл

Для сортування потрыбно використовувати меню Дії/Сортувати за потужністю (рис. 3.7):

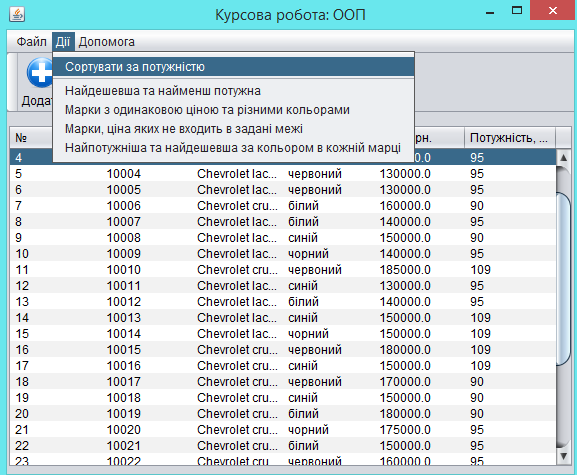


Рис. 3.7.Сортування за потужністю

Після такої операції таблиця з даними матиме вигляд (рис. 3.8):

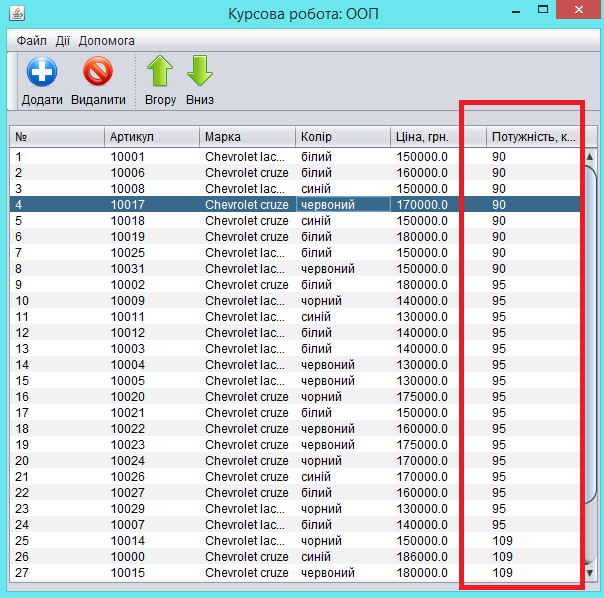


Рис. 3.8. Відсортована таблиця

Наступні пункти в меню дії: Найдешевша та найменш потужна (рис. 3.9); Марки з одинаковою ціною та різними кольорами (рис. 3.10); Марки, ціна яких не входить в задані межі (рис. 3.11); Найпотужніша та найдешевша за кольором в кожній марці (рис. 3.12).

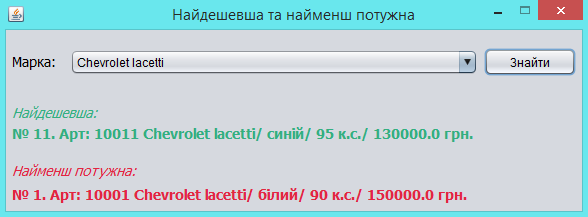


Рис. 3.9. Найдешевша та найменш потужна

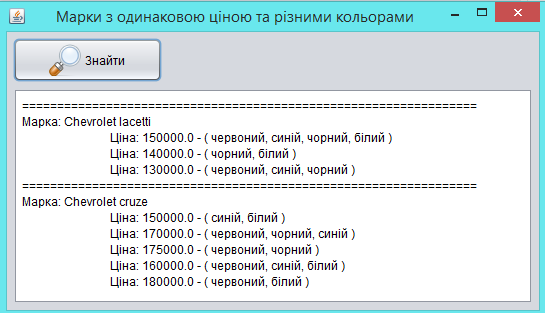


Рис. 3.10. Марки з одинаковою ціною та різними кольорами

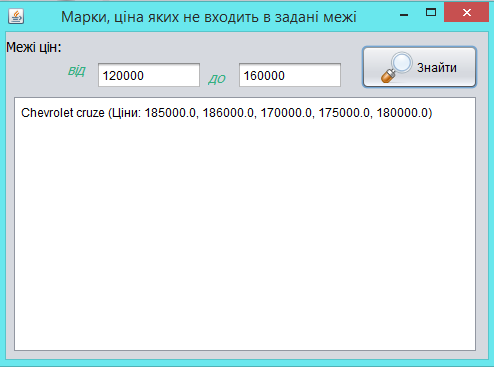


Рис. 3.11. Марки, ціна яких не входить в задані межі

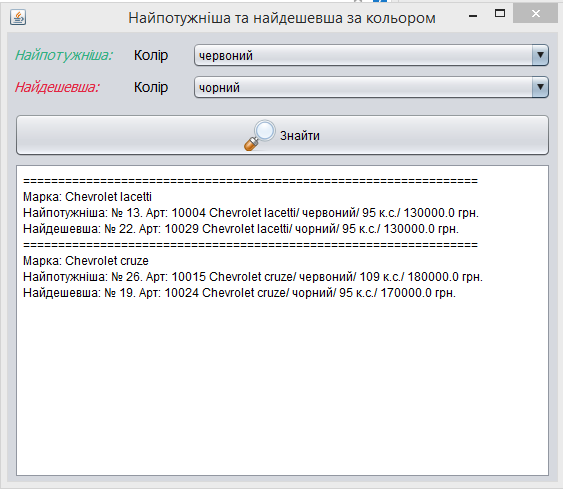


Рис. 3.12. Найпотужніша та найдешевша за кольором в кожній марці

**ВИСНОВКИ**

На даній курсовій роботі я поглибив свої знання у сфері ООП. Навчився будувавати графічний користувацький інтерфейс засобами Java. Вивчив типові структури даних в цій мові програмування, та механізми роботи з тестовими файлами.

**СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ**

1. Якобсон А., Буч Г., Рамбо Дж. Унифицированный процесс разработки программного обеспечения. – СПб.: Питер, 2002. – 498 с.:ил.

2. Буч Г., Рамбо Дж., Якобсон А Язык UML: Руководство пользователя. – М.: ДМК, 2000. – 356 с.: ил.

3. Мюллер Р.Дж. Базы данных и UML. Проектирование / Первод. с англ. Е. Молодцова. – М.: Издательство “Лори”,2002. – 432с.: ил.

4. Брюс Эккель. Философия Java. 4-е изд. - СПб.: Питер, 2009. - 640 с.

5. Кей С. Хорстманн, Гарри Корнелл Java 2, Т. 1, 2. М.: "Вильямс", 2007.

6. Yakov Fain. Java Programming. 24-Hour Trainer. – Indianapolis: Wiley Publishing, 2011.

7. Joshua Bloch. Effective Java. Second Edition. – Upper Saddle River, NJ, Boston, Indianapolis, San Francisco, New York, Toronto, Montreal, London, Munich, Paris, Madrid, Capetown, Sydney, Tokyo, Singapore, Mexico City: Addison-Wesley, 2008.

8. Анил Хемраджани. Гибкая разработка приложений на Java. – М.: Вильямс, 2008. – 352 с.

9. Ильдар Хабибуллин. Самоучитель Java. – СПб.: БХВ-Петербург, 2001. – 464 с.

**ДОДАТОК А. Лістинг програми**

**package com.taras\_grynchuk.oop;**

import com.taras\_grynchuk.oop.forms.MainWindow;

public class OOP {

public static void main(String[] args) {

MainWindow.main(args);

}

}

public class ListOfCars extends LinkedList<Car> implements Serializable {

//сортування за потужністю "методом вибору"

public void sort() {

for(int i = 0; i < size(); i++) {

// Знайдемо мінімальний елемент на

// проміжку індексів [i; size);

// спочатку його індекс рівний i

int minValueIndex = i;

// Перебираємо елементи, що залишилися на проміжку

for(int j = i + 1; j < size(); j++)

// Якщо елемент в позиції j менший

// елемента в позиції minValueIndex, то

// необхідно обновити значення індекса

if(get(j).getPower() < get(minValueIndex).getPower()) minValueIndex = j;

// Міняємо поточний елемент з мінімальним

Collections.swap(this, i, minValueIndex);

}

}

//список моделей без повторень

public String[] getArrayOfModels() {

HashSet<String> models = new HashSet<>();

for(Car c: this) models.add(c.getModel());

String[] arr = new String[models.size()];

models.toArray(arr);

return arr;

}

//список колборів без повторень

public String[] getArrayOfColors(String firstColor) {

HashSet<String> colors = new HashSet<>();

for(Car c: this) colors.add(c.getColor());

ArrayList<String> list = new ArrayList<>(colors);

if(firstColor.length() > 0) {

//ставимо вказаний колір на перше місце

int index = list.indexOf(firstColor);

if(index != -1) Collections.swap(list, index, 0);

}

String[] arr = new String[list.size()];

list.toArray(arr);

return arr;

}

public String[] getArrayOfColors() {

return this.getArrayOfColors("");

}

//найдешевша за вказаною моделлю

public int getMinPriceIndex(String model) {

return getMinPriceIndex(model, "");

}

//найдешевша за вказаною моделлю та кольором

public int getMinPriceIndex(String model, String color) {

float min = Float.MAX\_VALUE;

int index = -1;

for(int i = 0; i < size(); i++) {

Car c = get(i);

if(!model.toLowerCase().equals(c.getModel().toLowerCase())) continue;

if(color.length() > 0)

if(!color.toLowerCase().equals(c.getColor().toLowerCase())) continue;

float x = c.getPrice();

if(x < min) {

min = x;

index = i;

}

}

return index;

}

//найменш потужна

public int getMinPowerIndex(String model) {

int min = Integer.MAX\_VALUE;

int index = -1;

for(int i = 0; i < size(); i++) {

Car c = get(i);

if(!model.toLowerCase().equals(c.getModel().toLowerCase())) continue;

int x = c.getPower();

if(x < min) {

min = x;

index = i;

}

}

return index;

}

//марки з одинаковою ціною та різними кольорами

public String getSamePrice() {

StringBuilder s = new StringBuilder();

String[] models = getArrayOfModels();

//йдемо по моделях

for(String model: models) {

//створимо множину цін даної марки

HashSet<Float> prices = new HashSet<>();

for(Car car: this)

if(car.getModel().equals(model)) prices.add(car.getPrice());

String strPrices = "";

//йдемо по цінах

for(float price: prices) {

//створимо множину кольорів даної марки та даної ціни

HashSet<String> colors = new HashSet<>();

for(Car car: this)

if(car.getModel().equals(model)) //перевірка на модель

if(car.getPrice() == price) //перевірка на ціну

colors.add(car.getColor());

String strColors = "";

//якщо кольори різні (тобто їх к-сть в множині > 1)

if(colors.size() > 1) {

for(String color: colors) {

if(!strColors.equals("")) strColors += ", ";

strColors += color;

}

if(strPrices.length() > 0) strPrices += "\n";

strPrices += "\tЦіна: " + price + " - ( " + strColors + " )";

}

}

if(strPrices.length() > 0) {

if(s.length() != 0) s.append("\n");

s.append("=================================================================");

s.append("\nМарка: " + model);

s.append("\n" + strPrices);

}

}

return s.toString();

}

//Марки, ціна яких не входить в задані межі

public String getOutOfBoundsPrices(float priceFrom, float priceTo) {

StringBuilder s = new StringBuilder();

String[] models = getArrayOfModels();

//йдемо по моделях

for(String model: models) {

//створимо множину цін даної марки

HashSet<Float> prices = new HashSet<>();

for(Car car: this)

if(car.getModel().equals(model))

if(car.getPrice() > priceTo || car.getPrice() < priceFrom)

prices.add(car.getPrice());

String strPrices = "";

for(float price: prices) {

if(strPrices.length() > 0) strPrices += ", ";

strPrices += price;

}

if(strPrices.length() > 0) {

if(s.length() != 0) s.append("\n");

s.append(model + " (Ціни: " + strPrices + ")");

}

}

return s.toString();

}

//найпотужніша за вказаним кольором та моделлю

public int getMaxPowerIndex(String model, String color) {

int max = Integer.MIN\_VALUE;

int index = -1;

for(int i = 0; i < size(); i++) {

Car c = get(i);

if(!model.toLowerCase().equals(c.getModel().toLowerCase())) continue;

if(!color.toLowerCase().equals(c.getColor().toLowerCase())) continue;

int x = c.getPower();

if(x > max) {

max = x;

index = i;

}

}

return index;

}

//найпотужніша та найдешевша за вказаними кольорами у кожній марці

public String getMaxPowerAndMinPrice(String maxPowerColor, String minPriceColor) {

StringBuilder s = new StringBuilder();

String[] models = getArrayOfModels();

//йдемо по моделях

for(String model: models) {

if(s.length() != 0) s.append("\n");

s.append("=================================================================");

s.append("\nМарка: " + model);

s.append("\nНайпотужніша: ");

int index = getMaxPowerIndex(model, maxPowerColor);

if(index == -1) s.append("< не знайдено >");

else s.append("№ " + index + ". " + get(index));

s.append("\nНайдешевша: ");

index = getMinPriceIndex(model, minPriceColor);

if(index == -1) s.append("< не знайдено >");

else s.append("№ " + index + ". " + get(index));

}

return s.toString();

}

}

public class Car extends Goods implements Const, Serializable {

private static int count = 0; //загальна к-сть автомобілів

//атрибути автомобіля

private String color; //колір

private int power; //потужність

//------------------------------------------------

//методи встановлення/читання атрибутів

//загальна к-сть автомобілів

public static int getCount() {

return count;

}

public static void setCount(int newCount) {

count = newCount;

}

//артикул (не можна змінювати)

@Override

public int getArticle() {

return FIRST\_CAR\_ARTICLE + uid;

}

//марка

public String getModel() {

return getName();

}

public void setModel(String model) {

setName(model);

}

//колір

public String getColor() {

return color;

}

public void setColor(String color) {

this.color = color;

}

//потужність

public int getPower() {

return power;

}

public void setPower(int power) {

this.power = power;

}

//конструктори

public Car(String model, String color, float price, int power) {

this.uid = count++;

super.name = model;

this.color = color;

this.price = price;

this.power = power;

}

//пуста позицiя

public Car() {

this(EMPTY\_MODEL, EMPTY\_COLOR, 0f, 0);

}

//представлення об'єкта строкою

@Override

public String toString() {

return "Арт: " + getArticle() + " " + name + "/ " + color + "/ " + power + " к.с./ " + price + " грн.";

}

}

public abstract class Goods implements Serializable {

//атрибути товару

protected int uid; //унікальний ідентифікатор

protected String name; //Назва товару

protected float price; //ціна

//методи встановлення/читання атрибутів

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

//ціна

public float getPrice() {

return price;

}

public void setPrice(float price) {

this.price = price;

}

//отримання артмкулу

abstract int getArticle();

}

public interface Const {

int FIRST\_CAR\_ARTICLE = 10000; //початковий артикул автомобілів

String EMPTY\_MODEL = "Chevrolet"; //марка по-замовчуванню

String EMPTY\_COLOR = "<не вибрано>"; //колір по замовчуванню

String BLACK\_COLOR = "чорний";

String RED\_COLOR = "червоний";

}

**package com.taras\_grynchuk.oop.files;**

public class SerializableFiles {

//відкрити файл з списком машин

public static ListOfCars openFromSerFile() throws IOException, ClassNotFoundException {

FileInputStream fIn = null;

ObjectInputStream oIn = null;

fIn = new FileInputStream("ListOfCars.ser");

oIn = new ObjectInputStream(fIn);

ListOfCars list = new ListOfCars();

list = (ListOfCars) oIn.readObject();

if(oIn != null) {

oIn.close();

fIn.close();

}

Car.setCount(list.size());

return list;

}

//зберегти список машин у файл

public static void saveToSerFile(ListOfCars list) throws IOException, ClassNotFoundException {

FileOutputStream fOut = new FileOutputStream("ListOfCars.ser");

ObjectOutputStream oOut = new ObjectOutputStream(fOut);

oOut.writeObject(list);

oOut.flush();

oOut.close();

fOut.close();

}

}

public class TextFile {

public static void exportToTextFile(ListOfCars list, String fileName) throws IOException {

PrintWriter out = new PrintWriter(

new File(fileName).getAbsoluteFile());

for(Car car: list) {

char tab = (char) 9;

String s = "" +

car.getModel() + tab +

car.getColor() + tab +

car.getPrice() + tab +

car.getPower();

out.println(s);

}

out.close();

}

public static void importFromTextFile(ListOfCars list, String fileName) throws IOException, NumberFormatException, Exception {

BufferedReader in = new BufferedReader(new FileReader(fileName));

//new File(fileName).getAbsoluteFile()));

String s;

while((s = in.readLine()) != null) {

String[] a = s.split("\t");

Car car = new Car(a[0], a[1], Float.parseFloat(a[2]), Integer.parseInt(a[3]));

list.add(car);

}

in.close();

}

}

**package com.taras\_grynchuk.oop.forms;**

import com.taras\_grynchuk.oop.\*;

import static com.taras\_grynchuk.oop.files.SerializableFiles.\*;

import static com.taras\_grynchuk.oop.files.TextFile.\*;

import java.awt.\*;

import java.awt.Frame;

import java.io.File;

import java.io.FileFilter;

import java.util.\*;

import static java.util.Locale.filter;

import javax.swing.\*;

import javax.swing.event.\*;

import javax.swing.filechooser.FileNameExtensionFilter;

import javax.swing.table.\*;

public class MainWindow extends javax.swing.JFrame implements TableModelListener {

private MyTableModel myTableModel;

@Override

public void tableChanged(TableModelEvent e) {

}

class MyTableModel extends AbstractTableModel {

private final String[] colNames = {"№", "Артикул" ,"Марка", "Колiр", "Цiна, грн.", "Потужнiсть, к.с."};

@Override

public String getColumnName(int col) {

return colNames[col];

}

@Override

public boolean isCellEditable(int row, int col) {

return col > 1;

}

private ListOfCars myData;

public MyTableModel() {

try {

myData = openFromSerFile();

} catch(Exception e) {

myData = new ListOfCars();

myData.add(new Car());

}

}

@Override

public int getColumnCount() {

return colNames.length;

}

@Override

public int getRowCount() {

return myData.size();

}

@Override

public Object getValueAt(int row, int col) {

switch(col) {

case 0: return row + 1;

case 1: return myData.get(row).getArticle();

case 2: return myData.get(row).getModel();

case 3: return myData.get(row).getColor();

case 4: return myData.get(row).getPrice();

case 5: return myData.get(row).getPower();

default : return "";

}

}

@Override

public void setValueAt(Object value, int row, int col){

switch(col) {

case 2: myData.get(row).setModel(value.toString()); break;

case 3: myData.get(row).setColor(value.toString()); break;

case 4:

try {

myData.get(row).setPrice(Float.parseFloat(value.toString()));

} catch(NumberFormatException e) {

JOptionPane.showMessageDialog(MainWindow.this , "Неправильний формат числа !\n" +

e.getMessage(), "Помилка !", JOptionPane.ERROR\_MESSAGE);

} finally {

break;

}

case 5:

try {

myData.get(row).setPower(Integer.parseInt(value.toString()));

} catch(NumberFormatException e) {

JOptionPane.showMessageDialog(MainWindow.this , "Неправильний формат числа !\n" +

e.getMessage(), "Помилка !", JOptionPane.ERROR\_MESSAGE);

} finally {

break;

}

}

}

}

/\*\*

\* Creates new form MainWindow

\*/

public MainWindow() {

super("Курсова робота: ООП");

myTableModel = new MyTableModel();

initComponents();

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jScrollPane1 = new javax.swing.JScrollPane();

myTable = new javax.swing.JTable();

jToolBar1 = new javax.swing.JToolBar();

addButton = new javax.swing.JButton();

delButton = new javax.swing.JButton();

jSeparator1 = new javax.swing.JToolBar.Separator();

upButton = new javax.swing.JButton();

downButton = new javax.swing.JButton();

menuBar = new javax.swing.JMenuBar();

fileMenu = new javax.swing.JMenu();

openMenuItem = new javax.swing.JMenuItem();

saveMenuItem = new javax.swing.JMenuItem();

exitMenuItem = new javax.swing.JMenuItem();

editMenu = new javax.swing.JMenu();

sortMenuItem = new javax.swing.JMenuItem();

jSeparator2 = new javax.swing.JPopupMenu.Separator();

cheapestMenuItem = new javax.swing.JMenuItem();

samePriceMenuItem = new javax.swing.JMenuItem();

priceOutMenuItem = new javax.swing.JMenuItem();

minMaxMenuItem = new javax.swing.JMenuItem();

helpMenu = new javax.swing.JMenu();

contentsMenuItem = new javax.swing.JMenuItem();

aboutMenuItem = new javax.swing.JMenuItem();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

addWindowListener(new java.awt.event.WindowAdapter() {

public void windowClosing(java.awt.event.WindowEvent evt) {

formWindowClosing(evt);

}

});

myTable.setModel(myTableModel);

jScrollPane1.setViewportView(myTable);

jToolBar1.setRollover(true);

addButton.setIcon(new javax.swing.ImageIcon(getClass().getResource("/com/taras\_grynchuk/oop/forms/Add.png"))); // NOI18N

addButton.setText("Додати");

addButton.setFocusable(false);

addButton.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);

addButton.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);

addButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

addButtonActionPerformed(evt);

}

});

jToolBar1.add(addButton);

delButton.setIcon(new javax.swing.ImageIcon(getClass().getResource("/com/taras\_grynchuk/oop/forms/Delete.png"))); // NOI18N

delButton.setText("Видалити");

delButton.setFocusable(false);

delButton.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);

delButton.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);

delButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

delButtonActionPerformed(evt);

}

});

jToolBar1.add(delButton);

jToolBar1.add(jSeparator1);

upButton.setIcon(new javax.swing.ImageIcon(getClass().getResource("/com/taras\_grynchuk/oop/forms/Up.png"))); // NOI18N

upButton.setText("Вгору");

upButton.setFocusable(false);

upButton.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);

upButton.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);

upButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

upButtonActionPerformed(evt);

}

});

jToolBar1.add(upButton);

downButton.setIcon(new javax.swing.ImageIcon(getClass().getResource("/com/taras\_grynchuk/oop/forms/Down.png"))); // NOI18N

downButton.setText("Вниз");

downButton.setFocusable(false);

downButton.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);

downButton.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);

downButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

downButtonActionPerformed(evt);

}

});

jToolBar1.add(downButton);

fileMenu.setMnemonic('f');

fileMenu.setText("Файл");

openMenuItem.setMnemonic('o');

openMenuItem.setText("Імпорт ...");

openMenuItem.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

openMenuItemActionPerformed(evt);

}

});

fileMenu.add(openMenuItem);

saveMenuItem.setMnemonic('s');

saveMenuItem.setText("Експорт ...");

saveMenuItem.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

saveMenuItemActionPerformed(evt);

}

});

fileMenu.add(saveMenuItem);

exitMenuItem.setMnemonic('x');

exitMenuItem.setText("Вихід");

exitMenuItem.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

exitMenuItemActionPerformed(evt);

}

});

fileMenu.add(exitMenuItem);

menuBar.add(fileMenu);

editMenu.setMnemonic('e');

editMenu.setText("Дії");

sortMenuItem.setMnemonic('t');

sortMenuItem.setText("Сортувати за потужністю");

sortMenuItem.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

sortMenuItemActionPerformed(evt);

}

});

editMenu.add(sortMenuItem);

editMenu.add(jSeparator2);

cheapestMenuItem.setMnemonic('y');

cheapestMenuItem.setText("Найдешевша та найменш потужна");

cheapestMenuItem.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

cheapestMenuItemActionPerformed(evt);

}

});

editMenu.add(cheapestMenuItem);

samePriceMenuItem.setMnemonic('p');

samePriceMenuItem.setText("Марки з одинаковою ціною та різними кольорами");

samePriceMenuItem.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

samePriceMenuItemActionPerformed(evt);

}

});

editMenu.add(samePriceMenuItem);

priceOutMenuItem.setMnemonic('d');

priceOutMenuItem.setText("Марки, ціна яких не входить в задані межі");

priceOutMenuItem.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

priceOutMenuItemActionPerformed(evt);

}

});

editMenu.add(priceOutMenuItem);

minMaxMenuItem.setText("Найпотужніша та найдешевша за кольором в кожній марці");

minMaxMenuItem.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

minMaxMenuItemActionPerformed(evt);

}

});

editMenu.add(minMaxMenuItem);

menuBar.add(editMenu);

helpMenu.setMnemonic('h');

helpMenu.setText("Допомога");

contentsMenuItem.setMnemonic('c');

contentsMenuItem.setText("Довідка");

contentsMenuItem.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

contentsMenuItemActionPerformed(evt);

}

});

helpMenu.add(contentsMenuItem);

aboutMenuItem.setMnemonic('a');

aboutMenuItem.setText("Про програму");

aboutMenuItem.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

aboutMenuItemActionPerformed(evt);

}

});

helpMenu.add(aboutMenuItem);

menuBar.add(helpMenu);

setJMenuBar(menuBar);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jToolBar1, javax.swing.GroupLayout.DEFAULT\_SIZE, 592, Short.MAX\_VALUE)

.addComponent(jScrollPane1)

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addComponent(jToolBar1, javax.swing.GroupLayout.PREFERRED\_SIZE, 59, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT\_SIZE, 393, Short.MAX\_VALUE)

.addContainerGap())

);

pack();

}// </editor-fold>

private void exitMenuItemActionPerformed(java.awt.event.ActionEvent evt) {

formWindowClosing(null);

System.exit(0);

}

//додавання нового рядка

private void addButtonActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

myTableModel.myData.add(new Car());

myTable.revalidate();

}

//видалення рядка

private void delButtonActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

int[] rows = myTable.getSelectedRows();

for(int i = rows.length - 1; i >= 0; i--)

if(rows[i] < myTableModel.myData.size())

myTableModel.myData.remove(rows[i]);

myTable.revalidate();

}

//перемістити рядок вгору

private void upButtonActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

int row = myTable.getSelectedRow();

if((row >= myTableModel.myData.size()) || (row < 0)) return;

if(myTableModel.myData.size() < 2) return;

int newRow = row - 1;

if(newRow == -1) newRow = myTableModel.myData.size() - 1;

Collections.swap(myTableModel.myData, row, newRow);

myTable.revalidate();

myTable.repaint();

myTable.setRowSelectionInterval(newRow, newRow);

}

//перемістити рядок вниз

private void downButtonActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

int row = myTable.getSelectedRow();

if((row >= myTableModel.myData.size()) || (row < 0)) return;

if(myTableModel.myData.size() < 2) return;

int newRow = row + 1;

if(newRow == myTableModel.myData.size()) newRow = 0;

Collections.swap(myTableModel.myData, row, newRow);

myTable.revalidate();

myTable.repaint();

myTable.setRowSelectionInterval(newRow, newRow);

}

private void formWindowClosing(java.awt.event.WindowEvent evt) {

// TODO add your handling code here:

try {

saveToSerFile(myTableModel.myData);

} catch(Exception e) {

e.printStackTrace();

}

}

private void sortMenuItemActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

myTableModel.myData.sort();

myTable.revalidate();

myTable.repaint();

}

private void cheapestMenuItemActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

if(isDataEmpty()) return;

CheapestForm.open(myTableModel.myData);

}

private void samePriceMenuItemActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

if(isDataEmpty()) return;

SamePriceForm.open(myTableModel.myData);

}

private void priceOutMenuItemActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

if(isDataEmpty()) return;

PriceOutOfBoundsForm.open(myTableModel.myData);

}

private void minMaxMenuItemActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

if(isDataEmpty()) return;

MinMaxForm.open(myTableModel.myData);

}

private void saveMenuItemActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

JFileChooser saveFile = new JFileChooser();

FileNameExtensionFilter filter = new FileNameExtensionFilter(

"Текстові файли (\*.txt)", "txt");

saveFile.setFileFilter(filter);

if(saveFile.showSaveDialog(this) == JFileChooser.APPROVE\_OPTION) {

File file = saveFile.getSelectedFile();

String fileName = file.getAbsolutePath();

if(file.getName().toLowerCase().indexOf(".txt") == -1)

fileName += ".txt";

try {

exportToTextFile(myTableModel.myData, fileName);

JOptionPane.showMessageDialog(this, fileName, "Файл збережено !", JOptionPane.OK\_OPTION);

} catch(Exception e) {

JOptionPane.showMessageDialog(this, "Помилка збереження файлу !\n" +

fileName + "\n" +

e.getMessage(), "Помилка !", JOptionPane.ERROR\_MESSAGE);

}

}

}

private void openMenuItemActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

JFileChooser openFile = new JFileChooser();

FileNameExtensionFilter filter = new FileNameExtensionFilter(

"Текстові файли (\*.txt)", "txt");

openFile.setFileFilter(filter);

if(openFile.showOpenDialog(this) == JFileChooser.APPROVE\_OPTION) {

File file = openFile.getSelectedFile();

String fileName = file.getAbsolutePath();

try {

importFromTextFile(myTableModel.myData, fileName);

myTable.revalidate();

myTable.repaint();

JOptionPane.showMessageDialog(this, fileName, "Файл імпортовано !", JOptionPane.OK\_OPTION);

} catch(Exception e) {

JOptionPane.showMessageDialog(this, "Помилка відкриття файлу !\n" +

fileName + "\n", "Помилка !", JOptionPane.ERROR\_MESSAGE);

}

}

}

private void aboutMenuItemActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

AboutBox.main(null);

}

private void contentsMenuItemActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

Desktop dt = Desktop.getDesktop();

try {

dt.open(new File("ReadMe.html"));

} catch(Exception e) {

JOptionPane.showMessageDialog(this, "Помилка відкриття файлу довідки !", "Помилка !", JOptionPane.ERROR\_MESSAGE);

}

}

private boolean isDataEmpty() {

if(myTableModel.myData.size() == 0) {

JOptionPane.showMessageDialog(this, "Список пустий !", "Увага !", JOptionPane.WARNING\_MESSAGE);

return true;

}

return false;

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(MainWindow.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(MainWindow.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(MainWindow.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(MainWindow.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(new Runnable() {

@Override

public void run() {

new MainWindow().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JMenuItem aboutMenuItem;

private javax.swing.JButton addButton;

private javax.swing.JMenuItem cheapestMenuItem;

private javax.swing.JMenuItem contentsMenuItem;

private javax.swing.JButton delButton;

private javax.swing.JButton downButton;

private javax.swing.JMenu editMenu;

private javax.swing.JMenuItem exitMenuItem;

private javax.swing.JMenu fileMenu;

private javax.swing.JMenu helpMenu;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JToolBar.Separator jSeparator1;

private javax.swing.JPopupMenu.Separator jSeparator2;

private javax.swing.JToolBar jToolBar1;

private javax.swing.JMenuBar menuBar;

private javax.swing.JMenuItem minMaxMenuItem;

private javax.swing.JTable myTable;

private javax.swing.JMenuItem openMenuItem;

private javax.swing.JMenuItem priceOutMenuItem;

private javax.swing.JMenuItem samePriceMenuItem;

private javax.swing.JMenuItem saveMenuItem;

private javax.swing.JMenuItem sortMenuItem;

private javax.swing.JButton upButton;

// End of variables declaration

}

public class CheapestForm extends javax.swing.JFrame {

private ListOfCars cars;

/\*\*

\* Creates new form CheapestForm

\*/

public CheapestForm(ListOfCars cars) {

this.cars = cars;

initComponents();

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

modelLabel = new javax.swing.JLabel();

jComboBox1 = new javax.swing.JComboBox();

goButton = new javax.swing.JButton();

priceLabel = new javax.swing.JLabel();

priceLabelTitle = new javax.swing.JLabel();

powerLabelTitle = new javax.swing.JLabel();

powerLabel = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE\_ON\_CLOSE);

setTitle("Найдешевша та найменш потужна");

setAlwaysOnTop(true);

setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT\_CURSOR));

setLocationByPlatform(true);

setResizable(false);

modelLabel.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N

modelLabel.setText("Марка:");

modelLabel.setVerticalAlignment(javax.swing.SwingConstants.TOP);

jComboBox1.setModel(new DefaultComboBoxModel(cars.getArrayOfModels()));

goButton.setText("Знайти");

goButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

goButtonActionPerformed(evt);

}

});

priceLabel.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N

priceLabel.setForeground(new java.awt.Color(51, 175, 128));

priceLabel.setText("< ... >");

priceLabel.setVerticalAlignment(javax.swing.SwingConstants.TOP);

priceLabelTitle.setFont(new java.awt.Font("Tahoma", 2, 14)); // NOI18N

priceLabelTitle.setForeground(new java.awt.Color(51, 175, 128));

priceLabelTitle.setText("Найдешевша:");

priceLabelTitle.setVerticalAlignment(javax.swing.SwingConstants.TOP);

powerLabelTitle.setFont(new java.awt.Font("Tahoma", 2, 14)); // NOI18N

powerLabelTitle.setForeground(new java.awt.Color(227, 37, 66));

powerLabelTitle.setText("Найменш потужна:");

powerLabelTitle.setVerticalAlignment(javax.swing.SwingConstants.TOP);

powerLabelTitle.setMaximumSize(new java.awt.Dimension(85, 17));

powerLabelTitle.setMinimumSize(new java.awt.Dimension(85, 17));

powerLabelTitle.setPreferredSize(new java.awt.Dimension(85, 17));

powerLabel.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N

powerLabel.setForeground(new java.awt.Color(227, 37, 66));

powerLabel.setText("< ... >");

powerLabel.setVerticalAlignment(javax.swing.SwingConstants.TOP);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(priceLabel, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addGroup(layout.createSequentialGroup()

.addComponent(modelLabel, javax.swing.GroupLayout.PREFERRED\_SIZE, 52, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jComboBox1, 0, 398, Short.MAX\_VALUE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(goButton, javax.swing.GroupLayout.PREFERRED\_SIZE, 92, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addComponent(priceLabelTitle, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(powerLabelTitle, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(powerLabel, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addContainerGap())

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jComboBox1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(goButton, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(modelLabel, javax.swing.GroupLayout.DEFAULT\_SIZE, 35, Short.MAX\_VALUE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(priceLabelTitle, javax.swing.GroupLayout.PREFERRED\_SIZE, 20, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(1, 1, 1)

.addComponent(priceLabel, javax.swing.GroupLayout.PREFERRED\_SIZE, 25, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(powerLabelTitle, javax.swing.GroupLayout.PREFERRED\_SIZE, 23, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(1, 1, 1)

.addComponent(powerLabel, javax.swing.GroupLayout.PREFERRED\_SIZE, 25, javax.swing.GroupLayout.PREFERRED\_SIZE))

);

pack();

}// </editor-fold>

private void goButtonActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

int priceInd = cars.getMinPriceIndex(jComboBox1.getModel().getSelectedItem().toString());

priceLabel.setText((priceInd < 0 ? "< не знайдено >" :

"№ " + (1+priceInd) + ". " + cars.get(priceInd)));

int powerInd = cars.getMinPowerIndex(jComboBox1.getModel().getSelectedItem().toString());

powerLabel.setText((powerInd < 0 ? "< не знайдено >" :

"№ " + (1+powerInd) + ". " + cars.get(powerInd)));

}

public static void open(ListOfCars cars) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(CheapestForm.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(CheapestForm.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(CheapestForm.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(CheapestForm.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(new Runnable() {

@Override

public void run() {

new CheapestForm(cars).setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JButton goButton;

private javax.swing.JComboBox jComboBox1;

private javax.swing.JLabel modelLabel;

private javax.swing.JLabel powerLabel;

private javax.swing.JLabel powerLabelTitle;

private javax.swing.JLabel priceLabel;

private javax.swing.JLabel priceLabelTitle;

// End of variables declaration

}

public class MinMaxForm extends javax.swing.JFrame implements Const {

private ListOfCars cars;

/\*\*

\* Creates new form CheapestForm

\*/

public MinMaxForm(ListOfCars cars) {

this.cars = cars;

initComponents();

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

modelLabel = new javax.swing.JLabel();

jComboBox1 = new javax.swing.JComboBox();

goButton = new javax.swing.JButton();

maxPowerLabelTitle = new javax.swing.JLabel();

minPriceTitle = new javax.swing.JLabel();

modelLabel1 = new javax.swing.JLabel();

jComboBox2 = new javax.swing.JComboBox();

jScrollPane1 = new javax.swing.JScrollPane();

text = new javax.swing.JTextArea();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE\_ON\_CLOSE);

setTitle("Найпотужніша та найдешевша за кольором");

setAlwaysOnTop(true);

setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT\_CURSOR));

setLocationByPlatform(true);

modelLabel.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N

modelLabel.setText("Колір");

modelLabel.setVerticalAlignment(javax.swing.SwingConstants.TOP);

jComboBox1.setModel(new DefaultComboBoxModel(cars.getArrayOfColors(RED\_COLOR)));

goButton.setIcon(new javax.swing.ImageIcon(getClass().getResource("/com/taras\_grynchuk/oop/forms/Search.png"))); // NOI18N

goButton.setText("Знайти");

goButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

goButtonActionPerformed(evt);

}

});

maxPowerLabelTitle.setFont(new java.awt.Font("Tahoma", 2, 14)); // NOI18N

maxPowerLabelTitle.setForeground(new java.awt.Color(51, 175, 128));

maxPowerLabelTitle.setText("Найпотужніша:");

maxPowerLabelTitle.setVerticalAlignment(javax.swing.SwingConstants.TOP);

minPriceTitle.setFont(new java.awt.Font("Tahoma", 2, 14)); // NOI18N

minPriceTitle.setForeground(new java.awt.Color(227, 37, 66));

minPriceTitle.setText("Найдешевша:");

minPriceTitle.setVerticalAlignment(javax.swing.SwingConstants.TOP);

minPriceTitle.setMaximumSize(new java.awt.Dimension(85, 17));

minPriceTitle.setMinimumSize(new java.awt.Dimension(85, 17));

minPriceTitle.setPreferredSize(new java.awt.Dimension(85, 17));

modelLabel1.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N

modelLabel1.setText("Колір");

modelLabel1.setVerticalAlignment(javax.swing.SwingConstants.TOP);

jComboBox2.setModel(new DefaultComboBoxModel(cars.getArrayOfColors(BLACK\_COLOR)));

text.setColumns(20);

text.setRows(5);

jScrollPane1.setViewportView(text);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)

.addComponent(minPriceTitle, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(maxPowerLabelTitle, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.DEFAULT\_SIZE, 108, Short.MAX\_VALUE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(modelLabel, javax.swing.GroupLayout.PREFERRED\_SIZE, 52, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(modelLabel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 52, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jComboBox2, javax.swing.GroupLayout.PREFERRED\_SIZE, 359, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jComboBox1, javax.swing.GroupLayout.PREFERRED\_SIZE, 359, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addComponent(goButton, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jScrollPane1, javax.swing.GroupLayout.Alignment.TRAILING))

.addContainerGap())

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(9, 9, 9)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(maxPowerLabelTitle, javax.swing.GroupLayout.PREFERRED\_SIZE, 20, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(modelLabel)

.addComponent(jComboBox1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(minPriceTitle, javax.swing.GroupLayout.PREFERRED\_SIZE, 23, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(modelLabel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 23, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jComboBox2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(goButton)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT\_SIZE, 315, Short.MAX\_VALUE)

.addGap(3, 3, 3))

);

pack();

}// </editor-fold>

private void goButtonActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

text.setText(cars.getMaxPowerAndMinPrice(jComboBox1.getModel().getSelectedItem().toString(),

jComboBox2.getModel().getSelectedItem().toString()));

if(text.getText().length() == 0)

text.setText("Не знайдено жодного запису в таблиці !");

}

public static void open(ListOfCars cars) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(MinMaxForm.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(MinMaxForm.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(MinMaxForm.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(MinMaxForm.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(new Runnable() {

@Override

public void run() {

new MinMaxForm(cars).setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JButton goButton;

private javax.swing.JComboBox jComboBox1;

private javax.swing.JComboBox jComboBox2;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JLabel maxPowerLabelTitle;

private javax.swing.JLabel minPriceTitle;

private javax.swing.JLabel modelLabel;

private javax.swing.JLabel modelLabel1;

private javax.swing.JTextArea text;

// End of variables declaration

}

public class PriceOutOfBoundsForm extends javax.swing.JFrame {

private ListOfCars cars;

/\*\*

\* Creates new form PriceOutOfBoundsForm

\*/

public PriceOutOfBoundsForm(ListOfCars cars) {

this.cars = cars;

initComponents();

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

titleLabel = new javax.swing.JLabel();

labeTo = new javax.swing.JLabel();

textFrom = new javax.swing.JTextField();

labeFrom = new javax.swing.JLabel();

textTo = new javax.swing.JTextField();

goButton = new javax.swing.JButton();

jScrollPane1 = new javax.swing.JScrollPane();

text = new javax.swing.JTextArea();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE\_ON\_CLOSE);

setTitle("Марки, ціна яких не входить в задані межі");

setAlwaysOnTop(true);

titleLabel.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N

titleLabel.setText("Межі цін:");

titleLabel.setVerticalAlignment(javax.swing.SwingConstants.TOP);

labeTo.setFont(new java.awt.Font("Tahoma", 2, 14)); // NOI18N

labeTo.setForeground(new java.awt.Color(51, 175, 128));

labeTo.setText("до");

labeTo.setVerticalAlignment(javax.swing.SwingConstants.TOP);

textFrom.setText("100000");

labeFrom.setFont(new java.awt.Font("Tahoma", 2, 14)); // NOI18N

labeFrom.setForeground(new java.awt.Color(51, 175, 128));

labeFrom.setText("від");

labeFrom.setVerticalAlignment(javax.swing.SwingConstants.TOP);

textTo.setText("200000");

goButton.setIcon(new javax.swing.ImageIcon(getClass().getResource("/com/taras\_grynchuk/oop/forms/Search.png"))); // NOI18N

goButton.setText("Знайти");

goButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

goButtonActionPerformed(evt);

}

});

text.setColumns(20);

text.setRows(5);

jScrollPane1.setViewportView(text);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addGroup(javax.swing.GroupLayout.Alignment.LEADING, layout.createSequentialGroup()

.addContainerGap()

.addComponent(jScrollPane1))

.addGroup(javax.swing.GroupLayout.Alignment.LEADING, layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(titleLabel, javax.swing.GroupLayout.PREFERRED\_SIZE, 135, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGroup(layout.createSequentialGroup()

.addGap(61, 61, 61)

.addComponent(labeFrom, javax.swing.GroupLayout.PREFERRED\_SIZE, 23, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(textFrom, javax.swing.GroupLayout.PREFERRED\_SIZE, 106, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(labeTo, javax.swing.GroupLayout.PREFERRED\_SIZE, 23, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(textTo, javax.swing.GroupLayout.PREFERRED\_SIZE, 106, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addGap(18, 18, 18)

.addComponent(goButton, javax.swing.GroupLayout.PREFERRED\_SIZE, 117, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(0, 0, Short.MAX\_VALUE)))

.addGap(10, 10, 10))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(labeTo, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.PREFERRED\_SIZE, 20, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(textTo, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addComponent(titleLabel)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(textFrom, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(labeFrom, javax.swing.GroupLayout.PREFERRED\_SIZE, 20, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addComponent(goButton, javax.swing.GroupLayout.Alignment.TRAILING))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT\_SIZE, 258, Short.MAX\_VALUE)

.addContainerGap())

);

pack();

}// </editor-fold>

private void goButtonActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

float priceFrom, priceTo;

try {

priceFrom = Float.parseFloat(textFrom.getText());

priceTo = Float.parseFloat(textTo.getText());

} catch (NumberFormatException e) {

JOptionPane.showMessageDialog(this, "Неправильний формат числа !\n" +

e.getMessage(), "Помилка !", JOptionPane.ERROR\_MESSAGE);

return;

}

if(priceFrom > priceTo) {

text.setText("Ліва межа не може бути більшою за праву !");

return;

}

text.setText(cars.getOutOfBoundsPrices(priceFrom, priceTo));

if(text.getText().length() == 0)

text.setText("Не знайдено жодної марки ціна яклї не входить в задані межі !");

}

public static void open(ListOfCars cars) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(PriceOutOfBoundsForm.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(PriceOutOfBoundsForm.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(PriceOutOfBoundsForm.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(PriceOutOfBoundsForm.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(new Runnable() {

@Override

public void run() {

new PriceOutOfBoundsForm(cars).setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JButton goButton;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JLabel labeFrom;

private javax.swing.JLabel labeTo;

private javax.swing.JTextArea text;

private javax.swing.JTextField textFrom;

private javax.swing.JTextField textTo;

private javax.swing.JLabel titleLabel;

// End of variables declaration

}

public class SamePriceForm extends javax.swing.JFrame {

private ListOfCars cars;

/\*\*

\* Creates new form SamePriceForm

\*/

public SamePriceForm(ListOfCars cars) {

this.cars = cars;

initComponents();

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

goButton = new javax.swing.JButton();

jScrollPane1 = new javax.swing.JScrollPane();

text = new javax.swing.JTextArea();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE\_ON\_CLOSE);

setTitle("Марки з одинаковою ціною та різними кольорами");

setAlwaysOnTop(true);

goButton.setIcon(new javax.swing.ImageIcon(getClass().getResource("/com/taras\_grynchuk/oop/forms/Search.png"))); // NOI18N

goButton.setText("Знайти");

goButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

goButtonActionPerformed(evt);

}

});

text.setColumns(20);

text.setRows(5);

jScrollPane1.setViewportView(text);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jScrollPane1)

.addGroup(layout.createSequentialGroup()

.addComponent(goButton, javax.swing.GroupLayout.PREFERRED\_SIZE, 149, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(0, 405, Short.MAX\_VALUE)))

.addContainerGap())

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addComponent(goButton)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT\_SIZE, 274, Short.MAX\_VALUE)

.addContainerGap())

);

pack();

}// </editor-fold>

private void goButtonActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

text.setText(cars.getSamePrice());

if(text.getText().length() == 0)

text.setText("Не знайдено жодної марки з одинаковою ціною та різними кольорами !");

}

public static void open(ListOfCars cars) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(SamePriceForm.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(SamePriceForm.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(SamePriceForm.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(SamePriceForm.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(new Runnable() {

@Override

public void run() {

new SamePriceForm(cars).setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JButton goButton;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JTextArea text;

// End of variables declaration

}