­­­МИНОБРНАУКИ РОССИИ

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ

ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

«ЛЭТИ» им.В.И.УЛЬЯНОВА (ЛЕНИНА)

Кафедра вычислительной техники

Отчет по лабораторной работе №6

по дисциплине «Объектно-ориентированное программирование»

Тема: «**Обработка XML-документов**»

|  |  |
| --- | --- |
| Студент гр. 9307 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_Брызгалова Е.А |
|  |  |
| Преподаватель | \_\_\_\_\_\_\_\_\_\_\_\_\_\_Павловский М.Г. |

Санкт-Петербург

2021

**Цель работы**

Знакомство с технологией обработки XML-документов и файлов.

**Описание задания**

* Распечатки XML-файлов до загрузки данных в экранную форму и после их выгрузки.
* Скриншоты, иллюстрирующие процесс загрузки данных в XML файл и выгрузки из него.
* Текст документации, сгенерированный Javadoc.
* Фрагменты кода, отвечающие за сохранение и чтение данных из XML-файла.

**XML файлы до загрузки данных в экранную форму и после их выгрузки.**

На рис.1 представлен XML файл data.xml до загрузки в экранную форму.

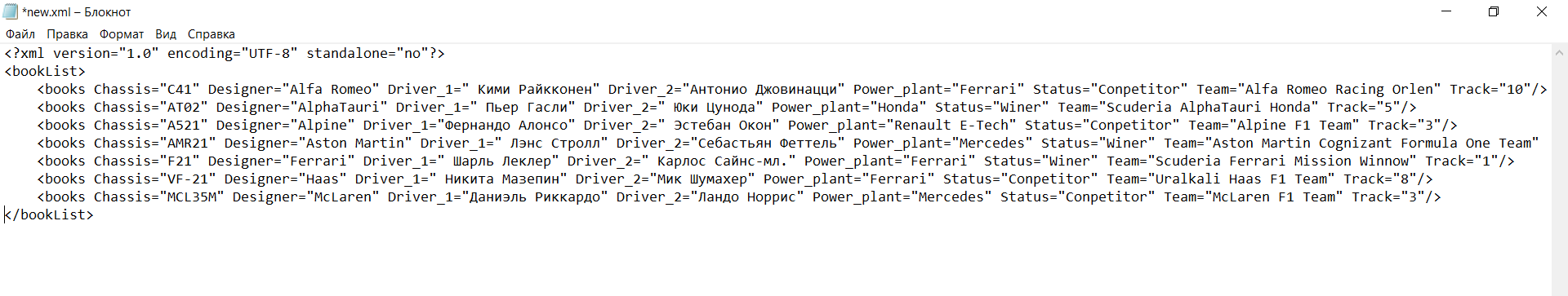


Рис.1 data.xml до загрузки данных

На рис.2 представлен процесс изменения данных.

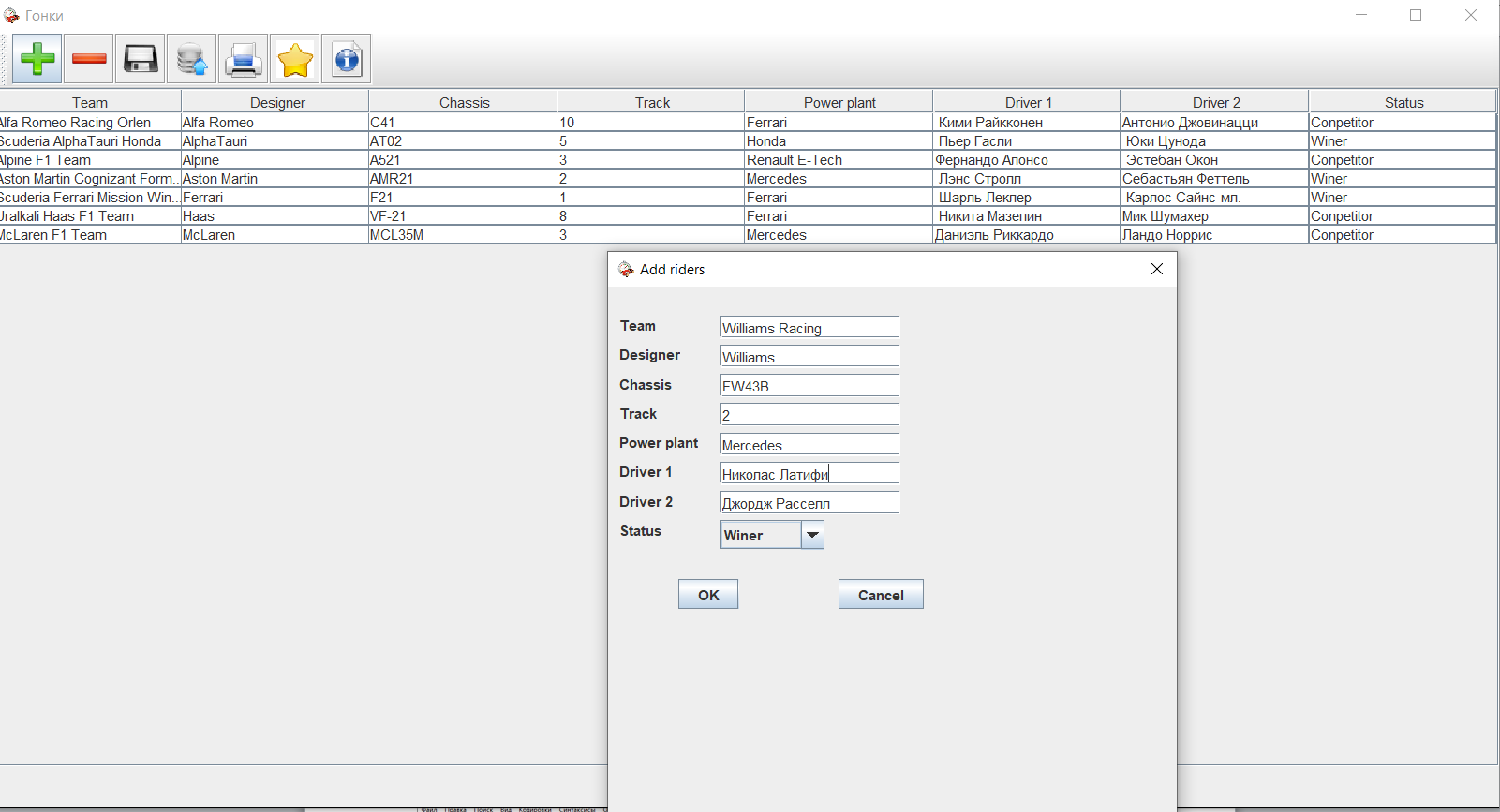


Рис.2 Добавление новой позиции

На рис.3 представлен изменённый файл data.xml

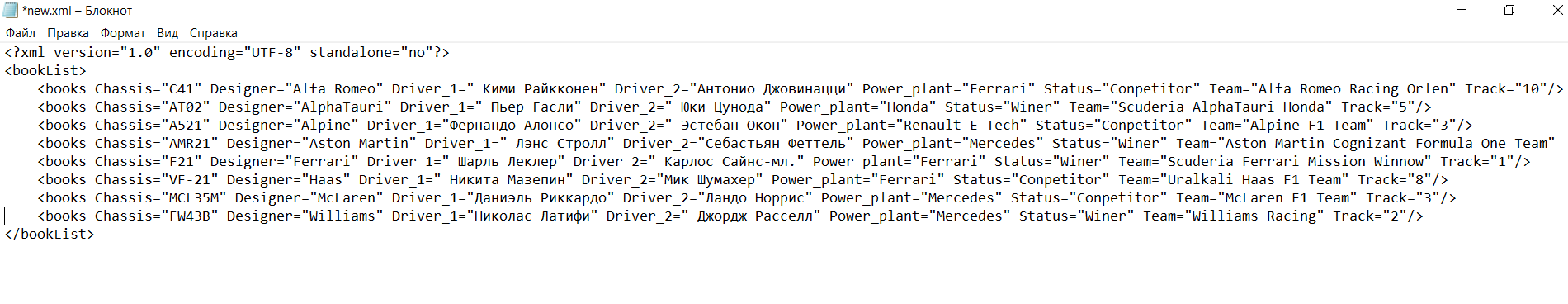


Рис.3 data.xml после выгрузки данных

**JavaDoc**

**Package** [edu.java.lab1](file:///C:\Eclips\work_space\First_lab\doc\edu\java\lab1\package-summary.html)

# Class First\_lab

java.lang.Object

edu.java.lab1.First\_lab

public class **First\_lab**

extends java.lang.Object

**Version:**

6.1

**Author:**

Bryzgalova Ekaterina 9307

## *Constructor Summary*

| **Constructors** | |
| --- | --- |
| **Constructor** | **Description** |
| [**First\_lab**](file:///C:\Eclips\work_space\First_lab\doc\edu\java\lab1\First_lab.html#%3Cinit%3E())() |  |

## *Method Summary*

All MethodsStatic MethodsInstance MethodsConcrete Methods

| **Modifier and Type** | **Method** | **Description** |
| --- | --- | --- |
| static void | [**main**](file:///C:\Eclips\work_space\First_lab\doc\edu\java\lab1\First_lab.html#main(java.lang.String%5B%5D))​(java.lang.String[] args) |  |
| void | [**show**](file:///C:\Eclips\work_space\First_lab\doc\edu\java\lab1\First_lab.html#show())() |  |

### Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## *Constructor Details*

### First\_lab

public First\_lab()

## *Method Details*

### show

public void show()

### main

public static void main​(java.lang.String[] args)

**Package** [edu.java.lab1](file:///C:\Eclips\work_space\First_lab\doc\edu\java\lab1\package-summary.html)

# Class Xml\_file

java.lang.Object

edu.java.lab1.Xml\_file

public class **Xml\_file**

extends java.lang.Object

Class for save or load data from XML file

## *Constructor Summary*

| **Constructors** | |
| --- | --- |
| **Constructor** | **Description** |
| [**Xml\_file**](file:///C:\Eclips\work_space\First_lab\doc\edu\java\lab1\Xml_file.html#%3Cinit%3E())() |  |

## *Method Summary*

All MethodsInstance MethodsConcrete Methods

| **Modifier and Type** | **Method** | **Description** |
| --- | --- | --- |
| void | [**xmlOpen**](file:///C:\Eclips\work_space\First_lab\doc\edu\java\lab1\Xml_file.html#xmlOpen(javax.swing.table.DefaultTableModel,javax.swing.JTable,javax.swing.JFrame))​(javax.swing.table.DefaultTableModel model, javax.swing.JTable table, javax.swing.JFrame frame) | method for open xml document |
| void | [**xmlSave**](file:///C:\Eclips\work_space\First_lab\doc\edu\java\lab1\Xml_file.html#xmlSave(javax.swing.table.DefaultTableModel,javax.swing.JTable,javax.swing.JFrame))​(javax.swing.table.DefaultTableModel model, javax.swing.JTable table, javax.swing.JFrame frame) | method for save xml document |

### Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## *Constructor Details*

### Xml\_file

public Xml\_file()

## *Method Details*

### xmlOpen

* + public void xmlOpen​(javax.swing.table.DefaultTableModel model,
  + javax.swing.JTable table,

javax.swing.JFrame frame)

method for open xml document

**Parameters:**

model - - model of table

table - - object type of JTable

frame - - object type of window

### xmlSave

* + public void xmlSave​(javax.swing.table.DefaultTableModel model,
  + javax.swing.JTable table,

javax.swing.JFrame frame) throws java.lang.Exception

method for save xml document

**Parameters:**

model - - model of table

table - - object type of JTable

frame - - object type of window

**Throws:**

java.lang.Exception

**Package** [edu.java.lab1](file:///C:\Eclips\work_space\First_lab\doc\edu\java\lab1\package-summary.html)

# Class Data

java.lang.Object

edu.java.lab1.Data

public class **Data**

extends java.lang.Object

Class for work to data

## *Constructor Summary*

| **Constructors** | |
| --- | --- |
| **Constructor** | **Description** |
| [**Data**](file:///C:\Eclips\work_space\First_lab\doc\edu\java\lab1\Data.html#%3Cinit%3E())() |  |

## *Method Summary*

All MethodsInstance MethodsConcrete Methods

| **Modifier and Type** | **Method** | **Description** |
| --- | --- | --- |
| void | [**Add**](file:///C:\Eclips\work_space\First_lab\doc\edu\java\lab1\Data.html#Add(javax.swing.table.DefaultTableModel,javax.swing.JFrame))​(javax.swing.table.DefaultTableModel model, javax.swing.JFrame frame) | Method for add new position in table |
| void | [**Delete**](file:///C:\Eclips\work_space\First_lab\doc\edu\java\lab1\Data.html#Delete(javax.swing.table.DefaultTableModel,javax.swing.JFrame))​(javax.swing.table.DefaultTableModel model, javax.swing.JFrame frame) | method for delete position |

### Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## *Constructor Details*

### Data

public Data()

## *Method Details*

### Add

* + public void Add​(javax.swing.table.DefaultTableModel model,

javax.swing.JFrame frame)

Method for add new position in table

**Parameters:**

model - - model of table

frame - - object type of window

### Delete

* + public void Delete​(javax.swing.table.DefaultTableModel model,

javax.swing.JFrame frame)

method for delete position

**Parameters:**

model - - model of table

frame - - object type of window

**Код программы**

**Fist\_lab.java**

package edu.java.lab1;

// Подключение графических библиотек

import javax.swing.\*;

import javax.swing.table.DefaultTableModel;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.util.Vector;

import javax.swing.JOptionPane;

import java.awt.print.PrinterException;

import java.text.MessageFormat;

import javax.swing.table.TableModel;

import javax.swing.table.TableRowSorter;

/\*\*

\* @author Bryzgalova Ekaterina 9307

\* @version 6.1

\*/

public class First\_lab {

// Объявления графических компонентов

private JFrame bookList;

private DefaultTableModel model;

private JButton delete;

private JButton load;

private JButton save;

private JButton add;

private JButton print;

private JButton info;

private JButton winners;

private JToolBar toolBar;

private JTable books;

private JTextField Name;

private JButton filter;

//Создание окна

public void show() {

// Создание окна

bookList = new JFrame("Гонки");

bookList.setSize(1300, 700);

bookList.setLocation(100, 100);

ImageIcon icon = new ImageIcon("./pictures\_OOP/car1.png");

bookList.setIconImage(icon.getImage());

bookList.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

// Создание кнопок и прикрепление иконок

add = new JButton(new ImageIcon("./pictures\_OOP/add.png"));

save = new JButton(new ImageIcon("./pictures\_OOP/save.png"));

load = new JButton(new ImageIcon("./pictures\_OOP/load.png"));

print = new JButton(new ImageIcon("./pictures\_OOP/print.png"));

delete = new JButton(new ImageIcon("./pictures\_OOP/delete.png"));

info = new JButton(new ImageIcon("./pictures\_OOP/info.png"));

winners=new JButton(new ImageIcon("./pictures\_OOP/rating.png"));

filter = new JButton("Search");

// Настройка подсказок для кнопок

add.setToolTipText("Edd position");

load.setToolTipText("Download the list of riders");

save.setToolTipText("Save the list of riders");

print.setToolTipText("Print");

winners.setToolTipText("List of winners");

delete.setToolTipText("Delete a position");

info.setToolTipText("Program information");

// Добавление кнопок на панель инструментов

toolBar = new JToolBar("Toolbar");

toolBar.add(add);

toolBar.add(delete);

toolBar.add(save);

toolBar.add(load);

toolBar.add(print);

toolBar.add(winners);

toolBar.add(info);

// Размещение панели инструментов

bookList.setLayout(new BorderLayout());

bookList.add(toolBar, BorderLayout.NORTH);

// Создание таблицы с данными

final String [] column = {"Team", "Designer", "Chassis","Track","Power plant","Driver 1", "Driver 2","Status"};

Vector<String> columns = new Vector<>();

for(int i = 0;i < column.length;i++) {

columns.add(column[i]);

}

Vector <Vector<String>> data = new Vector<>();

model= new DefaultTableModel(data, columns);

books = new JTable(model);

final TableRowSorter <TableModel> sorter = new TableRowSorter<>(model);

books.setRowSorter(sorter);

JScrollPane scroll = new JScrollPane(books);

scroll = new JScrollPane(books);

Data d = new Data();

Xml\_file fl = new Xml\_file();

// Обработка нажатия кнопки загрузки базы данных

load.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

fl.xmlOpen(model,books,bookList);

}

});

// Обработка нажатия кнопки поиск

filter.addActionListener (new ActionListener()

{

public void actionPerformed (ActionEvent event)

{

String text =Name.getText();

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

sorter.setRowFilter(null);

} else {

sorter.setRowFilter(RowFilter.regexFilter(text));

}

}

});

winners.addActionListener (new ActionListener()

{

int f=1;

public void actionPerformed (ActionEvent event)

{

if(f==1) {

sorter.setRowFilter(RowFilter.regexFilter("Winer"));

f=0;

}

else {

sorter.setRowFilter(null);

f=1;

}

}

});

// Обработка нажатия кнопки информация о программе

info.addActionListener (new ActionListener()

{

public void actionPerformed (ActionEvent event)

{

JOptionPane.showMessageDialog (bookList, "Author: Bryzgalova E.A.\nVersion 5.1");

}

});

// Обработка нажатия кнопки добавления новой позиции

add.addActionListener(new ActionListener()

{

public void actionPerformed(ActionEvent e)

{

d.Add(model,bookList);

}

});

// Обработка нажатия кнопки удаление

delete.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

d.Delete(model,bookList);

}

});

//Обработка нажатия кнопки сохранения

save.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

//db.SaveFile(model);

try {

fl.xmlSave(model,books,bookList);

} catch (Exception exception) {

exception.printStackTrace();

}

}

});

print.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

try {

MessageFormat headerFormat = new MessageFormat("Page {0}");

MessageFormat footerFormat = new MessageFormat("- {0} -");

books.print(JTable.PrintMode.FIT\_WIDTH, headerFormat, footerFormat);

} catch (PrinterException pe) {

System.err.println("Error printing: " + pe.getMessage());

}

}

});

/////////////////////////////////////////////////////////////////////////

// Размещение таблицы с данными

bookList.add(scroll, BorderLayout.CENTER);

// Подготовка компонентов поиска

Name = new JTextField("Team");

// Добавление компонентов на панель

JPanel filterPanel = new JPanel();

filterPanel.add(Name);

filterPanel.add(filter);

// Размещение панели поиска внизу окна

bookList.add(filterPanel, BorderLayout.SOUTH);

books.setEnabled(false);

// Визуализация экранной формы

bookList.setVisible(true);

}

public static void main(String[] args) {

// Создание и отображение экранной формы

new First\_lab().show();

}

}

**Data.java**

package edu.java.lab1;

import javax.swing.\*;

import javax.swing.table.DefaultTableModel;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.BufferedReader;

import java.io.IOException;

import java.nio.file.Files;

import java.nio.file.Paths;

import java.util.Vector;

import javax.swing.JOptionPane;

import static javax.swing.WindowConstants.DISPOSE\_ON\_CLOSE;

/\*\*

\* Class for work to data

\*/

public class Data {

/\*\*

\* Method for add new position in table

\* @param model - model of table

\* @param frame - object type of window

\*/

public void Add(DefaultTableModel model,JFrame frame) {

JDialog window = new JDialog(frame,"Add riders");

window.setModal(true);

window.setSize(500,500);

window.setLocation(300,300);

Container contentPane = window.getContentPane();

SpringLayout layout = new SpringLayout();

contentPane.setLayout(layout);

Component Team = new JLabel("Team");

JTextField tTeam = new JTextField(15);

contentPane.add(Team);

contentPane.add(tTeam);

layout.putConstraint(SpringLayout.WEST , Team, 10,

SpringLayout.WEST , contentPane);

layout.putConstraint(SpringLayout.NORTH, Team, 25,

SpringLayout.NORTH, contentPane);

layout.putConstraint(SpringLayout.NORTH, tTeam, 25,

SpringLayout.NORTH, contentPane);

layout.putConstraint(SpringLayout.WEST , tTeam, 55,

SpringLayout.EAST , Team);

Component Designer = new JLabel("Designer");

JTextField tDesigner = new JTextField(15);

layout.putConstraint(SpringLayout.WEST , Designer, 10,

SpringLayout.WEST , contentPane);

layout.putConstraint(SpringLayout.NORTH, Designer, 50,

SpringLayout.NORTH, contentPane);

layout.putConstraint(SpringLayout.NORTH, tDesigner, 50,

SpringLayout.NORTH, contentPane);

layout.putConstraint(SpringLayout.WEST , tDesigner, 34,

SpringLayout.EAST , Designer);

contentPane.add(Designer);

contentPane.add(tDesigner);

Component Chassis = new JLabel("Chassis");

JTextField tChassis = new JTextField(15);

layout.putConstraint(SpringLayout.WEST , Chassis, 10,

SpringLayout.WEST , contentPane);

layout.putConstraint(SpringLayout.NORTH, Chassis, 75,

SpringLayout.NORTH, contentPane);

layout.putConstraint(SpringLayout.NORTH, tChassis, 75,

SpringLayout.NORTH, contentPane);

layout.putConstraint(SpringLayout.WEST , tChassis, 41,

SpringLayout.EAST ,Chassis);

contentPane.add(Chassis);

contentPane.add(tChassis);

Component Track = new JLabel("Track");

JTextField tTrack = new JTextField(15);

layout.putConstraint(SpringLayout.WEST , Track, 10,

SpringLayout.WEST , contentPane);

layout.putConstraint(SpringLayout.NORTH, Track, 100,

SpringLayout.NORTH, contentPane);

layout.putConstraint(SpringLayout.NORTH, tTrack, 100,

SpringLayout.NORTH, contentPane);

layout.putConstraint(SpringLayout.WEST , tTrack, 54,

SpringLayout.EAST , Track );

contentPane.add(Track);

contentPane.add(tTrack);

Component Power\_plant = new JLabel("Power plant");

JTextField tPower\_plant = new JTextField(15);

layout.putConstraint(SpringLayout.WEST , Power\_plant, 10,

SpringLayout.WEST , contentPane);

layout.putConstraint(SpringLayout.NORTH, Power\_plant, 125,

SpringLayout.NORTH, contentPane);

layout.putConstraint(SpringLayout.NORTH, tPower\_plant, 125,

SpringLayout.NORTH, contentPane);

layout.putConstraint(SpringLayout.WEST , tPower\_plant, 19,

SpringLayout.EAST , Power\_plant );

contentPane.add(Power\_plant);

contentPane.add(tPower\_plant);

Component Driver\_1 = new JLabel("Driver 1");

JTextField tDriver\_1 = new JTextField(15);

layout.putConstraint(SpringLayout.WEST , Driver\_1, 10,

SpringLayout.WEST , contentPane);

layout.putConstraint(SpringLayout.NORTH, Driver\_1, 150,

SpringLayout.NORTH, contentPane);

layout.putConstraint(SpringLayout.NORTH, tDriver\_1, 150,

SpringLayout.NORTH, contentPane);

layout.putConstraint(SpringLayout.WEST , tDriver\_1, 40,

SpringLayout.EAST ,Driver\_1);

contentPane.add(Driver\_1);

contentPane.add(tDriver\_1);

Component Driver\_2 = new JLabel("Driver 2");

JTextField tDriver\_2 = new JTextField(15);

layout.putConstraint(SpringLayout.WEST , Driver\_2, 10,

SpringLayout.WEST , contentPane);

layout.putConstraint(SpringLayout.NORTH, Driver\_2, 175,

SpringLayout.NORTH, contentPane);

layout.putConstraint(SpringLayout.NORTH, tDriver\_2, 175,

SpringLayout.NORTH, contentPane);

layout.putConstraint(SpringLayout.WEST , tDriver\_2, 40,

SpringLayout.EAST , Driver\_2);

contentPane.add(Driver\_2);

contentPane.add( tDriver\_2 );

Component Status = new JLabel("Status");

String S[] = {"Winer","Conpetitor"};

JComboBox<String> comboAvailability = new JComboBox<>(S);

layout.putConstraint(SpringLayout.WEST ,Status, 10,

SpringLayout.WEST , contentPane);

layout.putConstraint(SpringLayout.NORTH, Status, 200,

SpringLayout.NORTH, contentPane);

layout.putConstraint(SpringLayout.NORTH, comboAvailability, 200,

SpringLayout.NORTH, contentPane);

layout.putConstraint(SpringLayout.WEST , comboAvailability, 50,

SpringLayout.EAST , Status);

contentPane.add(Status);

contentPane.add(comboAvailability);

JButton ok = new JButton("OK");

JButton cancel = new JButton("Cancel");

layout.putConstraint(SpringLayout.WEST , ok, 60,

SpringLayout.WEST , contentPane);

layout.putConstraint(SpringLayout.NORTH, ok, 250,

SpringLayout.NORTH, contentPane);

layout.putConstraint(SpringLayout.NORTH, cancel, 250,

SpringLayout.NORTH, contentPane);

layout.putConstraint(SpringLayout.WEST , cancel, 85,

SpringLayout.EAST , ok );

contentPane.add(ok);

contentPane.add(cancel);

ok.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

Vector <String> s = new Vector<>();

s.add(tTeam.getText());

s.add(tDesigner.getText());

s.add(tChassis.getText());

s.add(tTrack.getText());

s.add(tPower\_plant.getText());

s.add(tDriver\_1.getText());

s.add(tDriver\_2.getText());

s.add(S[comboAvailability.getSelectedIndex()]);

int count = 0;

for(String s1:s) {

if((s1.length() > 0) && (!s1.equals(" "))) {

count++;

}

}

if(count == 8) {

model.addRow(s);

window.dispose();

}

else {

JOptionPane.showMessageDialog (frame,"Not all fields are filled in!");

}

}

});

cancel.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

int n = JOptionPane.showConfirmDialog(window,"Are you sure you want to close the window?","Message",JOptionPane.YES\_NO\_OPTION);

if(n == JOptionPane.YES\_OPTION) {

window.dispose();

}

}

});

window.setDefaultCloseOperation(DISPOSE\_ON\_CLOSE);

window.setVisible(true);

}

/\*\*

\* method for delete position

\* @param model - model of table

\* @param frame - object type of window

\*/

public void Delete(DefaultTableModel model, JFrame frame) {

try {

model.removeRow(model.getRowCount()-1);

}

catch (IndexOutOfBoundsException ev) {

JOptionPane.showMessageDialog (frame,"A mistake! The table is empty!","Error", JOptionPane.ERROR\_MESSAGE);

}

}

}

**Xml\_file**

package edu.java.lab1;

import org.w3c.dom.\*;

import org.xml.sax.SAXException;

import javax.swing.\*;

import javax.swing.table.DefaultTableModel;

import java.io.FileOutputStream;

import java.lang.Exception;

import javax.xml.parsers.DocumentBuilder;

import javax.xml.parsers.DocumentBuilderFactory;

import javax.xml.transform.\*;

import javax.xml.transform.dom.DOMSource;

import javax.xml.transform.stream.StreamResult;

import java.awt.\*;

import java.io.File;

import java.io.IOException;

/\*\*

\* Class for save or load data from XML file

\*/

public class Xml\_file {

private static String fileNameOpen;

private static String fileNameSave;

private static String tags[] = {"Team", "Designer", "Chassis","Track","Power\_plant","Driver\_1", "Driver\_2","Status"};

/\*\*

\* method getting document for open

\*/

private static Document getDocumentOpen() throws Exception {

try {

DocumentBuilderFactory f = DocumentBuilderFactory.newInstance();

DocumentBuilder builder = f.newDocumentBuilder();

return builder.parse(new File(fileNameOpen));

} catch (Exception exception) {

throw new Exception("XML parsing error!");

}

}

/\*\*

\* method getting document for save

\*/

private static Document getDocumentSave() throws Exception {

try {

DocumentBuilderFactory f = DocumentBuilderFactory.newInstance();

DocumentBuilder builder = f.newDocumentBuilder();

return builder.newDocument();

} catch (Exception exception) {

throw new Exception("XML parsing error!");

}

}

/\*\*

\* method for open xml document

\* @param model - model of table

\* @param table - object type of JTable

\* @param frame - object type of window

\*/

public void xmlOpen(DefaultTableModel model, JTable table,JFrame frame) {

FileDialog openXml = new FileDialog(frame,"Load File",FileDialog.LOAD);

openXml.setFile(".xml");

openXml.setVisible(true);

fileNameOpen = openXml.getDirectory() + openXml.getFile();

if(fileNameOpen == null) return;

model = (DefaultTableModel) table.getModel();

model.setRowCount(0);

try {

Document doc = getDocumentOpen();

doc.getDocumentElement().normalize();

NodeList list = doc.getElementsByTagName("books");

String s[] = new String[tags.length];

for(int i = 0;i < list.getLength();i++) {

Node elem = list.item(i);

NamedNodeMap atr = elem.getAttributes();

for(int j = 0;j < tags.length;j++) {

s[j] = atr.getNamedItem(tags[j]).getNodeValue();

}

model.addRow(s);

}

}catch(SAXException e){

e.printStackTrace();

}catch(Exception e){

e.printStackTrace();

}

}

/\*\*

\* method for save xml document

\* @param model - model of table

\* @param table - object type of JTable

\* @param frame - object type of window

\* @throws Exception

\*/

public void xmlSave(DefaultTableModel model, JTable table,JFrame frame) throws Exception {

FileDialog saveXml = new FileDialog(frame,"Save File",FileDialog.SAVE);

saveXml.setFile("\*.xml");

saveXml.setVisible(true);

fileNameSave = saveXml.getDirectory() + saveXml.getFile();

if(fileNameSave == null) return;

Document doc = getDocumentSave();

Node booklist = doc.createElement("bookList");

doc.appendChild(booklist);

for(int i = 0; i < model.getRowCount(); i++){

Element books = doc.createElement("books");

booklist.appendChild(books);

for(int j = 0;j < tags.length;j++) {

books.setAttribute(tags[j],(String) model.getValueAt(i,j));

}

}

try{

Transformer trans= TransformerFactory.newInstance().newTransformer();

trans.setOutputProperty(OutputKeys.METHOD, "xml");

trans.setOutputProperty(OutputKeys.INDENT, "yes");

trans.transform(new DOMSource(doc), new StreamResult(new FileOutputStream(fileNameSave)));

}catch(TransformerConfigurationException e){

e.printStackTrace();

}catch(TransformerException e){

e.printStackTrace();

}catch(IOException e){

e.printStackTrace();

}

}

}