­­­МИНОБРНАУКИ РОССИИ

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ

ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

«ЛЭТИ» им.В.И.УЛЬЯНОВА (ЛЕНИНА)

Кафедра вычислительной техники

Отчет по лабораторной работе №8

по дисциплине «Объектно-ориентированное программирование»

Тема: «**Организация многопоточных приложений**»

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

Санкт-Петербург

2021

Оглавление

[Цель работы 3](#_Toc72767655)

[Описание задания 3](#_Toc72767656)

[Проверка работоспособности программы 3](#_Toc72767657)

[Фрагмент кода. Потоки 8](#_Toc72767658)

[Вывод 9](#_Toc72767659)

# Цель работы

Знакомство с правилами и классами построения параллельных приложений в языке Java

# Описание задания

* Скриншоты созданных отчётов.
* Текст документации, сгенерированный Javadoc.
* Фрагменты кода, отвечающие за создание отчётов в формате PDF и HTML.

# Проверка работоспособности программы

При запуске программы активизируются три потока:

* Загрузка данных из файла data.xml
* Редактирование данных
* Создание отчётов в формате pdf и html

На рис.1 представлен скриншот исходного файла data.xml до изменения.

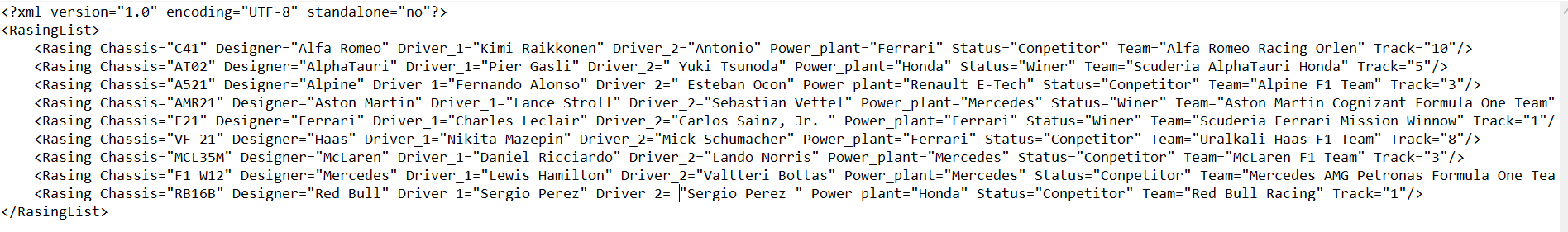


Рис.1

На рис.2 и рис.3 представлены скриншоты отчётов в формате pdf и html после изменения данных.

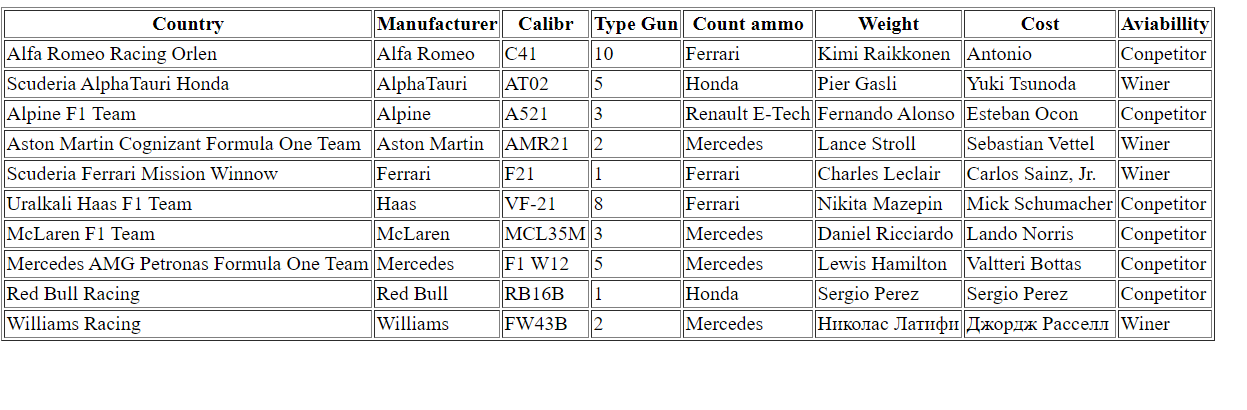


Рис.2 Отчёт в формате html

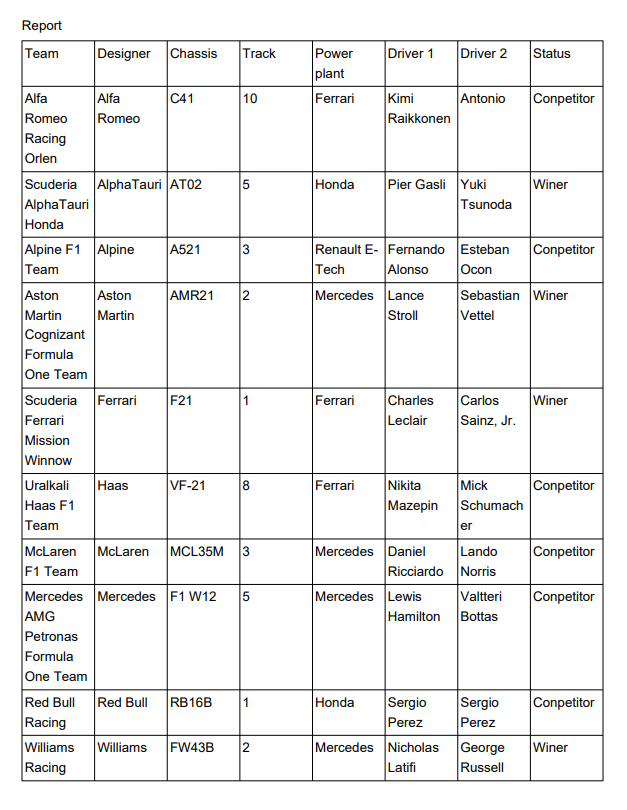


Рис.3 Отчёт в формате pdf

Class Main

java.lang.Object

Main

public class **Main**

extends java.lang.Object

*Constructor Summary*

| **Constructors** | |
| --- | --- |
| **Constructor** | **Description** |
| [**Main**](file:///C:\Eclips\work_space\lab8\doc\Main.html#%3Cinit%3E())() |  |

*Method Summary*

All MethodsStatic MethodsConcrete Methods

| **Modifier and Type** | **Method** | **Description** |
| --- | --- | --- |
| static void | [**main**](file:///C:\Eclips\work_space\lab8\doc\Main.html#main(java.lang.String%5B%5D))​(java.lang.String[] args) |  |

Methods inherited from class java.lang.Object

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

*Constructor Details*

Main

public Main()

*Method Details*

main

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

Class Window

java.lang.Object

Window

public class **Window**

extends java.lang.Object

class for create main window

*Field Summary*

| **Fields** | | |
| --- | --- | --- |
| **Modifier and Type** | **Field** | **Description** |
| static java.lang.Object | [**shared**](file:///C:\Eclips\work_space\lab8\doc\Window.html#shared) |  |

*Constructor Summary*

| **Constructors** | |
| --- | --- |
| **Constructor** | **Description** |
| [**Window**](file:///C:\Eclips\work_space\lab8\doc\Window.html#%3Cinit%3E())() | Constructor class Window |

*Method Summary*

Methods inherited from class java.lang.Object

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

*Field Details*

shared

public static final java.lang.Object shared

*Constructor Details*

Window

public Window()

Constructor class Window

Class Data

java.lang.Object

Data

public class **Data**

extends java.lang.Object

Class for work to data

*Constructor Summary*

| **Constructors** | |
| --- | --- |
| **Constructor** | **Description** |
| [**Data**](file:///C:\Eclips\work_space\lab8\doc\Data.html#%3Cinit%3E())() |  |

*Method Summary*

All MethodsInstance MethodsConcrete Methods

| **Modifier and Type** | **Method** | **Description** |
| --- | --- | --- |
| void | [**Add**](file:///C:\Eclips\work_space\lab8\doc\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\lab8\doc\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

Class Report

java.lang.Object

Report

public class **Report**

extends java.lang.Object

*Constructor Summary*

| **Constructors** | |
| --- | --- |
| **Constructor** | **Description** |
| [**Report**](file:///C:\Eclips\work_space\lab8\doc\Report.html#%3Cinit%3E())() |  |

*Method Summary*

Methods inherited from class java.lang.Object

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

*Constructor Details*

Report

public Report()

Class Xml\_file

java.lang.Object

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\lab8\doc\Xml_file.html#%3Cinit%3E())() |  |

*Method Summary*

All MethodsInstance MethodsConcrete Methods

| **Modifier and Type** | **Method** | **Description** |
| --- | --- | --- |
| void | [**xmlOpen**](file:///C:\Eclips\work_space\lab8\doc\Xml_file.html#xmlOpen(javax.swing.table.DefaultTableModel))​(javax.swing.table.DefaultTableModel model) | method for open xml document |
| void | [**xmlSave**](file:///C:\Eclips\work_space\lab8\doc\Xml_file.html#xmlSave(javax.swing.table.DefaultTableModel))​(javax.swing.table.DefaultTableModel model) | 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)

method for open xml document

**Parameters:**

model - - model of table

xmlSave

public void xmlSave​(javax.swing.table.DefaultTableModel model) throws java.lang.Exception

method for save xml document

**Parameters:**

model - - model of table

**Throws:**

java.lang.Exception

# Фрагмент кода. Потоки

**class** tthread **extends** Thread {

**private** **int** type;

**public** tthread(**int** i) {

type = i;

}

**public** **void** run() {

**if** (type == 1) {

**synchronized** (***shared***) {

**try** {

***shared***.wait();

} **catch** (InterruptedException e) {

e.printStackTrace();

}

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

fl.xmlOpen(model);

System.***out***.println("potok 1");

}

}

**if** (type == 2) {

**synchronized** (***shared***) {

***shared***.notifyAll();

**try** {

***shared***.wait();

} **catch** (InterruptedException e) {

e.printStackTrace();

}

model.addRow(**new** String[]{"Williams Racing","Williams","FW43B","2","Mercedes","Nicholas Latifi", "George Russell","Winer"});

System.***out***.println("potok 2");

***shared***.notifyAll();

}

}

**if** (type == 3) {

**synchronized** (***shared***) {

***shared***.notifyAll();

**try** {

***shared***.wait();

} **catch** (InterruptedException e) {

e.printStackTrace();

}

Report rp = **new** Report();

rp.createHTML(model);

**try** {

rp.createPDF(model);

} **catch** (FileNotFoundException e) {

e.printStackTrace();

}

System.***out***.println("potok 3");

}

}

}

}

**new** tthread(1).start();

**new** tthread(2).start();

**new** tthread(3).start();

# Вывод

В ходе лабораторной работы были полученны навыки работы с потоками, была создана многопоточная программа.