**МИНОБРНАУКИ РОССИИ**

федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Санкт-Петербургский государственный технологический институт (технический университет)»**

**(СПбГТИ(ТУ))**

Факультет **Информационных технологий и управления**

Кафедра **Систем автоматизированного проектирования и управления**

Дисциплина **Технология разработки программного обеспечения**

**Лабораторная работа №4**

# **Тема: организация множественных прикладных сред с использованием системы виртуальных машин**

# **разработка интерфейса к бд**

Захаренко К.А.

Комарова И. А.

Группа № 4811М

Проверили:

Корниенко И.Г.

Федин А.К.

Санкт-Петербург

2018

**Цель работы**

В данной работе предлагается получить навыки создания систем, поддерживающих идею множественных прикладных сред, воспользовавшись программным обеспечением системы виртуальных машин.

**Результаты по организации виртуальной машины**

Загрузки



Рисунок 1 – Главное окно программы Oracle VM VirtualBox

Создание виртуальной машины



Рисунок 2 – Выбор ОС

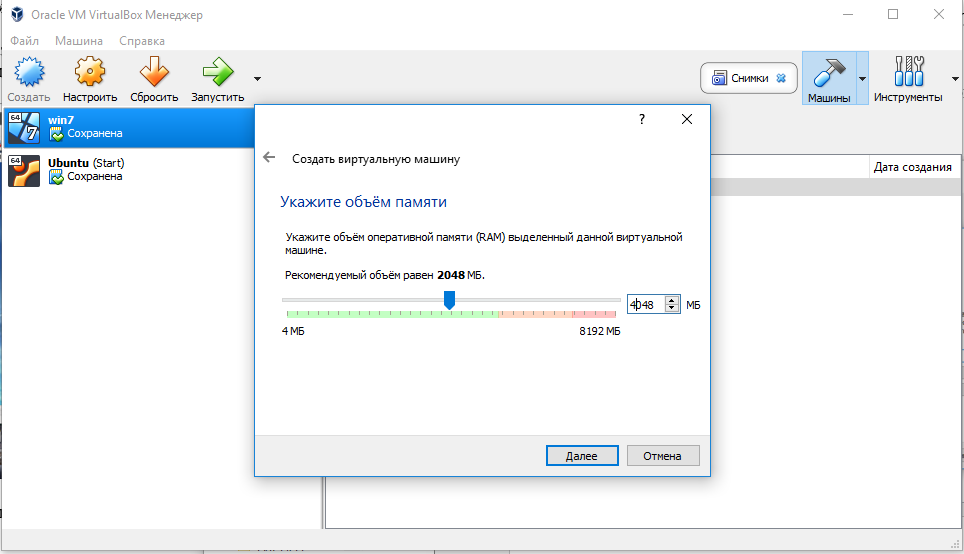


Рисунок 3 – Выбор объема памяти

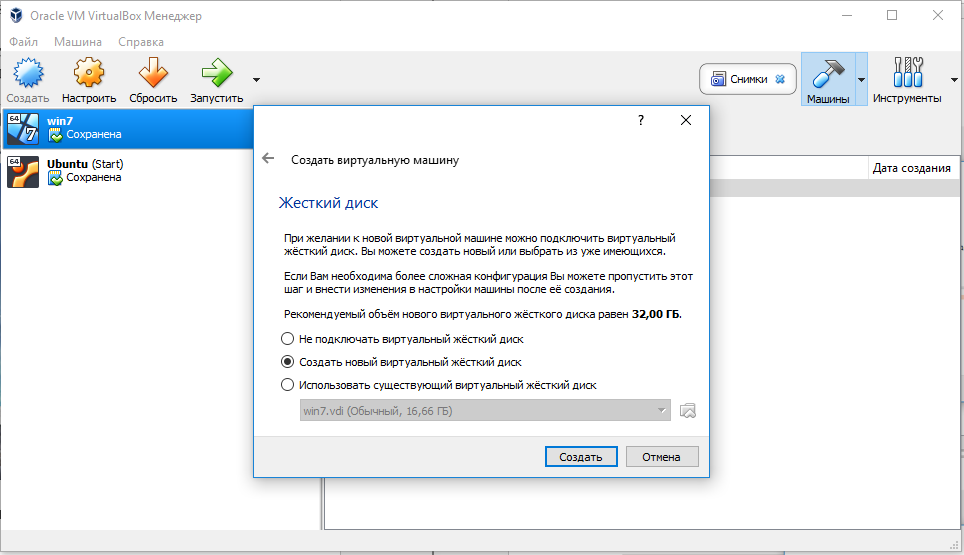


Рисунок 4 – Подключение ЖД

Рисунок 5 – Тип ЖД



Рисунок 6 – Формат хранения



Рисунок 7 – Имя и размер ЖД

Инсталляция гостевой операционной системы

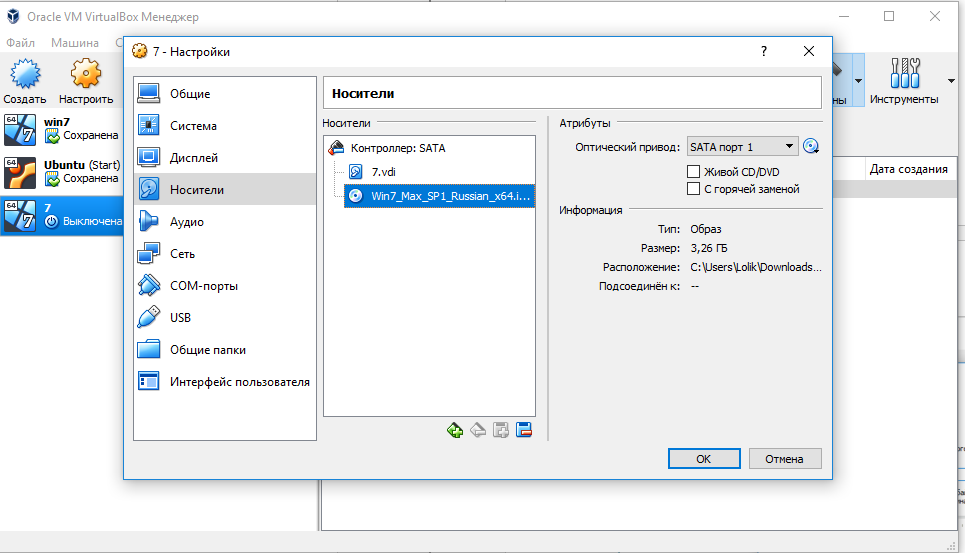


Рисунок 8 – Настройки виртуальной машины



Рисунок 9 – Установка ОС на виртуальную машину

Работа на виртуальной машине



Рисунок 10 – Работа на виртуальной машине

**Листинг разработанной программы для связи с БД**

public partial class Form10 : Form

{

public class ComboBoxKostyl

{

public string id;

public string name;

public override string ToString()

{

return name;

}

}

public static Form10 instance;

public string id\_man;

bool updateVars = false;

public Form10()

{

// InitializeComponent();

instance = this;

InitializeComponent();

updateVars = true;

SQLiteCommand command = new SQLiteCommand("select \* from type\_eq", Form1.instance.BD);

DbDataReader reader = command.ExecuteReader();

while (reader.Read())

{

typ.Items.Add(new ComboBoxKostyl() { id = reader["ID\_type"].ToString(), name = reader["Name\_type"].ToString() });

// typ.Items.Add(new { id\_type = reader["ID\_type"].ToString(), name\_eq = reader["Name\_type"].ToString() });

}

DataSet ds = new DataSet();

string sql = "SELECT ID\_eq, Name\_eq FROM equipment";

dataGridView1.RowPostPaint += dataGridView1\_RowPostPaint;

SQLiteCommand comm = new SQLiteCommand(sql, Form1.instance.BD);

using (SQLiteDataReader read = comm.ExecuteReader())

{

while (read.Read())

{

dataGridView1.Rows.Add(new object[] {

read.GetValue(1),

});

dataGridView1.Rows[dataGridView1.Rows.Count - 1].Tag = read.GetValue(0);

}

}

dataGridView1.SelectionChanged += new EventHandler(dataGridView1\_SelectionChanged);

updateVars = false;

}

private void dataGridView1\_RowPostPaint(object sender, DataGridViewRowPostPaintEventArgs e)

{

var grid = sender as DataGridView;

var centerFormat = new StringFormat()

{

Alignment = StringAlignment.Center,

LineAlignment = StringAlignment.Center

};

var headerBounds = new Rectangle(e.RowBounds.Left, e.RowBounds.Top, grid.RowHeadersWidth, e.RowBounds.Height);

}

private void dataGridView1\_SelectionChanged(object sender, EventArgs e)

{

updateVars = true;

/\* string sql = String.Format("SELECT manufactory.Name\_man, manufactory.Length\_man, manufactory.Width\_man, manufactory.Hight\_man, " +

"manufactory.Performance, manufactory.Energy, manufactory.Cost, manufactory.Path\_man, " +

"Group\_Concat(extrusion\_line.Name, ', ') FROM composition\_man JOIN manufactory ON " +

"manufactory.ID\_man=composition\_man.ID\_man JOIN extrusion\_line ON composition\_man.ID\_line=extrusion\_line.ID\_line"); \*/

string sql = "SELECT equipment.Name\_eq, type\_eq.Name\_type, equipment.Length\_eq, equipment.Width\_eq, " +

"equipment.Hight\_eq, equipment.Cost, equipment.Path\_eq, equipment.link\_eq, equipment.ID\_eq FROM equipment JOIN type\_eq ON equipment.ID\_type = type\_eq.ID\_type";

dataGridView1.SelectedRows[0].Tag.ToString();

sql += " WHERE equipment.ID\_eq = " + dataGridView1.SelectedRows[0].Tag.ToString();

DataSet ds = new DataSet();

SQLiteCommand comm = new SQLiteCommand(sql, Form1.instance.BD);

using (SQLiteDataReader read = comm.ExecuteReader())

{

while (read.Read())

{

Name\_eq.Text = read.GetString(0);

//typ.SelectedValue = read.GetValue(1);

//typ.SelectedItem = read.GetValue(1);

string desiredType = read.GetValue(1).ToString();

foreach (ComboBoxKostyl item in typ.Items)

{

if (item.name.Equals(desiredType))

{

typ.SelectedItem = item;

break;

}

}

Length.Value = (decimal)read.GetDouble(2);

Width.Value = (decimal)read.GetDouble(3);

High.Value = (decimal)read.GetDouble(4);

cost.Value = (decimal)read.GetDouble(5);

path.Text = read.GetString(6);

model.Text = read.GetString(7);

}

}

SQLiteCommand CMD = Form1.instance.BD.CreateCommand();

//ПРОИЗВОДИТЕЛЬНОСТЬ

CMD.CommandText = "SELECT equipment\_param\_eq.value AS PER FROM param\_eq JOIN equipment\_param\_eq ON " +

"param\_eq.ID\_param=equipment\_param\_eq.ID\_param WHERE equipment\_param\_eq.ID\_param =1 AND equipment\_param\_eq.ID\_eq = " + "'" + dataGridView1.SelectedRows[0].Tag.ToString() + "'";

//string p = CMD.ExecuteScalar().ToString();

//per.Value = Decimal.Parse(p);

//per.Value = Decimal.Parse(CMD.ExecuteScalar().ToString());

per.Value = Convert.ToDecimal(CMD.ExecuteScalar().ToString());

//label2.Text = per.Value.ToString();

// per.Value = (decimal)CMD.ExecuteScalar();

//ЭНЕРГОПОТРЕБ

CMD.CommandText = "SELECT equipment\_param\_eq.value AS PER FROM param\_eq JOIN equipment\_param\_eq ON " +

"param\_eq.ID\_param=equipment\_param\_eq.ID\_param WHERE equipment\_param\_eq.ID\_param =2 AND equipment\_param\_eq.ID\_eq = " + "'" + dataGridView1.SelectedRows[0].Tag.ToString() + "'";

// energy.Text = CMD.ExecuteScalar().ToString();

energy.Value = Convert.ToDecimal(CMD.ExecuteScalar().ToString());

updateVars = false;

}

private void Name\_eq\_TextChanged(object sender, EventArgs e)

{

if (!updateVars)

{

SQLiteCommand CMD = Form1.instance.BD.CreateCommand();

CMD.CommandText = "UPDATE equipment SET Name\_eq = " + "'" + Name\_eq.Text + "'" + " WHERE ID\_eq = " + dataGridView1.SelectedRows[0].Tag.ToString();

CMD.ExecuteNonQuery();

dataGridView1.SelectedRows[0].Cells[0].Value = Name\_eq.Text;

}

}

private void typ\_SelectedIndexChanged(object sender, EventArgs e)

{

if (!updateVars)

{

SQLiteCommand CMD = Form1.instance.BD.CreateCommand();

CMD.CommandText = "UPDATE equipment SET ID\_type= " + "'" + (typ.SelectedItem as ComboBoxKostyl).id + "'" + " WHERE ID\_eq = " + dataGridView1.SelectedRows[0].Tag.ToString();

CMD.ExecuteNonQuery();

}

}

private void Length\_ValueChanged(object sender, EventArgs e)

{

if (!updateVars)

{

SQLiteCommand CMD = Form1.instance.BD.CreateCommand();

CMD.CommandText = "UPDATE equipment SET Length\_eq = " + "'" + Length.Value.ToString().ToLower().Replace(",", ".") + "'" + " WHERE ID\_eq = " + dataGridView1.SelectedRows[0].Tag.ToString();

CMD.ExecuteNonQuery();

}

}

private void Width\_ValueChanged(object sender, EventArgs e)

{

if (!updateVars)

{

SQLiteCommand CMD = Form1.instance.BD.CreateCommand();

CMD.CommandText = "UPDATE equipment SET Width\_eq = " + "'" + Width.Value.ToString().ToLower().Replace(",", ".") + "'" + " WHERE ID\_eq = " + dataGridView1.SelectedRows[0].Tag.ToString();

CMD.ExecuteNonQuery();

}

}

private void High\_ValueChanged(object sender, EventArgs e)

{

if (!updateVars)

{

SQLiteCommand CMD = Form1.instance.BD.CreateCommand();

CMD.CommandText = "UPDATE equipment SET Hight\_eq = " + "'" + High.Value.ToString().ToLower().Replace(",", ".") + "'" + " WHERE ID\_eq = " + dataGridView1.SelectedRows[0].Tag.ToString();

CMD.ExecuteNonQuery();

}

}

private void cost\_ValueChanged(object sender, EventArgs e)

{

if (!updateVars)

{

SQLiteCommand CMD = Form1.instance.BD.CreateCommand();

CMD.CommandText = "UPDATE equipment SET Cost = " + "'" + cost.Value.ToString().ToLower().Replace(",", ".") + "'" + " WHERE ID\_eq = " + dataGridView1.SelectedRows[0].Tag.ToString();

CMD.ExecuteNonQuery();

}

}

private void per\_ValueChanged(object sender, EventArgs e)

{

if (!updateVars)

{

SQLiteCommand CMD = Form1.instance.BD.CreateCommand();

//ПРОИЗВОДИТЕЛЬНОСТЬ

CMD.CommandText = "UPDATE equipment\_param\_eq SET value = " + "'" + per.Value.ToString().ToLower().Replace(",", ".") + "'" + " WHERE ID\_param=1 AND ID\_eq=" + dataGridView1.SelectedRows[0].Tag.ToString() ;

CMD.ExecuteNonQuery();

// per.Text = CMD.ExecuteScalar().ToString();

}

}

private void energy\_ValueChanged(object sender, EventArgs e)

{

if (!updateVars)

{

SQLiteCommand CMD = Form1.instance.BD.CreateCommand();

//energyyy

CMD.CommandText = "UPDATE equipment\_param\_eq SET value = " + "'" + energy.Value.ToString().ToLower().Replace(",", ".") + "'" + " WHERE ID\_param=2 AND ID\_eq=" + dataGridView1.SelectedRows[0].Tag.ToString();

CMD.ExecuteNonQuery();

// per.Text = CMD.ExecuteScalar().ToString();

}

}

private void path\_TextChanged(object sender, EventArgs e)

{

if (!updateVars)

{

SQLiteCommand CMD = Form1.instance.BD.CreateCommand();

CMD.CommandText = "UPDATE equipment SET Path\_eq = " + "'" + path.Text + "'" + " WHERE ID\_eq = " + dataGridView1.SelectedRows[0].Tag.ToString();

CMD.ExecuteNonQuery();

}

}

private void model\_TextChanged(object sender, EventArgs e)

{

if (!updateVars)

{

SQLiteCommand CMD = Form1.instance.BD.CreateCommand();

CMD.CommandText = "UPDATE equipment SET link\_eq = " + "'" + model.Text + "'" + " WHERE ID\_eq = " + dataGridView1.SelectedRows[0].Tag.ToString();

CMD.ExecuteNonQuery();

}

}

private void Form10\_FormClosing(object sender, FormClosingEventArgs e)

{

//костыль, запись всех ид в лист

string sql = "SELECT ID\_eq FROM equipment"; ;

SQLiteCommand comm = new SQLiteCommand(sql, Form1.instance.BD);

List<string> IDeq = new List<string>();

using (SQLiteDataReader read = comm.ExecuteReader())

{

while (read.Read())

{

IDeq.Add(read[0].ToString());

}

}

for (int l = 0; l < IDeq.Count; l++)

{

SQLiteCommand CMD1 = Form1.instance.BD.CreateCommand();

CMD1.CommandText = "SELECT Length\_eq FROM equipment " + "'" + IDeq[l] + "'";

SQLiteCommand CMD2 = Form1.instance.BD.CreateCommand();

CMD2.CommandText = "SELECT Width\_eq FROM equipment " + "'" + IDeq[l] + "'";

SQLiteCommand CMD3 = Form1.instance.BD.CreateCommand();

CMD3.CommandText = "SELECT Hight\_eq FROM equipment " + "'" + IDeq[l] + "'";

SQLiteCommand CMD4 = Form1.instance.BD.CreateCommand();

CMD4.CommandText = "SELECT Cost FROM equipment " + "'" + IDeq[l] + "'";

SQLiteCommand CMD5 = Form1.instance.BD.CreateCommand();

CMD5.CommandText = "SELECT equipment\_param\_eq.value AS PER FROM param\_eq JOIN equipment\_param\_eq ON " +

"param\_eq.ID\_param=equipment\_param\_eq.ID\_param WHERE equipment\_param\_eq.ID\_param =1 AND equipment\_param\_eq.ID\_eq = " + "'" + IDeq[l] + "'";

SQLiteCommand CMD6 = Form1.instance.BD.CreateCommand();

CMD6.CommandText = "SELECT equipment\_param\_eq.value AS PER FROM param\_eq JOIN equipment\_param\_eq ON " +

"param\_eq.ID\_param=equipment\_param\_eq.ID\_param WHERE equipment\_param\_eq.ID\_param =2 AND equipment\_param\_eq.ID\_eq = " + "'" + IDeq[l] + "'";

if (Convert.ToDouble(CMD1.ExecuteScalar()) == 0 || Convert.ToDouble(CMD2.ExecuteScalar()) == 0 || Convert.ToDouble(CMD3.ExecuteScalar()) == 0 ||

Convert.ToDouble(CMD4.ExecuteScalar()) == 0 || Convert.ToDouble(CMD5.ExecuteScalar()) == 0 || Convert.ToDouble(CMD6.ExecuteScalar()) == 0 )

{

MessageBox.Show("Значения не должны быть нулевыми!", "ОШИБКА!", MessageBoxButtons.OK, MessageBoxIcon.Error);

e.Cancel = true;

break;

}

SQLiteCommand CMD7 = Form1.instance.BD.CreateCommand();

CMD7.CommandText = "SELECT Name\_eq FROM equipment WHERE ID\_eq= " + "'" + IDeq[l] + "'";

SQLiteCommand CMD8 = Form1.instance.BD.CreateCommand();

CMD8.CommandText = "SELECT Path\_eq FROM equipment WHERE ID\_eq= " + "'" + IDeq[l] + "'";

SQLiteCommand CMD9 = Form1.instance.BD.CreateCommand();

CMD9.CommandText = "SELECT link\_eq FROM equipment WHERE ID\_eq= " + "'" + IDeq[l] + "'";

if (CMD7.ExecuteScalar().ToString() == "" || CMD8.ExecuteScalar().ToString() == "" || CMD9.ExecuteScalar().ToString() == "")

{

MessageBox.Show("Заполните все поля!", "ОШИБКА!", MessageBoxButtons.OK, MessageBoxIcon.Error);

e.Cancel = true;

break;

}

}

}

private void button2\_Click(object sender, EventArgs e)

{

dataGridView1.Rows.Add("Оборудование");

SQLiteCommand CMD = Form1.instance.BD.CreateCommand();

CMD.CommandText = "INSERT INTO equipment(Name\_eq, Length\_eq, Width\_eq, Hight\_eq, Cost, " +

"Path\_eq, link\_eq, ID\_type) VALUES ('Оборудование', '0', '0', '0', '0', '/path', 'link', 1)";

CMD.ExecuteNonQuery();

DataSet ds = new DataSet();

string sql = "SELECT MAX(ID\_eq) FROM equipment";

SQLiteCommand comm = new SQLiteCommand(sql, Form1.instance.BD);

var id = comm.ExecuteScalar();

dataGridView1.Rows[dataGridView1.Rows.Count - 1].Tag = id;

CMD.CommandText = "INSERT INTO equipment\_param\_eq(value, ID\_param, ID\_units, ID\_eq) VALUES ('0', 1, 1, " + "'" + id + "'"+ ")" ;

CMD.ExecuteNonQuery();

CMD.CommandText = "INSERT INTO equipment\_param\_eq(value, ID\_param, ID\_units, ID\_eq) VALUES ('0', 2, 1, " + "'" + id + "'" + ")";

CMD.ExecuteNonQuery();

}

private void button3\_Click(object sender, EventArgs e)

{

SQLiteCommand CMD = Form1.instance.BD.CreateCommand();

CMD.CommandText = "DELETE FROM equipment WHERE ID\_eq = " + dataGridView1.Rows[dataGridView1.SelectedRows[0].Index].Tag;

CMD.ExecuteNonQuery();

CMD.CommandText = "DELETE FROM composition\_line WHERE ID\_eq = " + dataGridView1.Rows[dataGridView1.SelectedRows[0].Index].Tag;

CMD.ExecuteNonQuery();

CMD.CommandText = "DELETE FROM equipment\_param\_eq WHERE ID\_eq = " + dataGridView1.Rows[dataGridView1.SelectedRows[0].Index].Tag;

CMD.ExecuteNonQuery();

dataGridView1.Rows.RemoveAt(dataGridView1.SelectedRows[0].Index);

}

private void Form10\_FormClosed(object sender, FormClosedEventArgs e)

{

Form1.instance.updatePerformance();

Form1.instance.update\_cex();

Form1.instance.updateTable2();

Form1.instance.updateTable();

Form1.instance.updateTable3();

}

}