МИНОБРНАУКИ РОССИИ

федеральное государственное бюджетное образовательное учреждение   
высшего образования   
«Санкт-Петербургский государственный технологический институт  
(технический университет)»  
СПбГТИ(ТУ)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| УГНС | | 09.00.00 | Информатика и вычислительная техника | | |
| Направление подготовки | | 09.04.01 | Информатика и вычислительная техника | | |
| Направленность (профиль) | |  | Системы автоматизированного проектирования | | |
| Форма обучения | |  | очная | | |
|  | |  |  | | |
| Факультет | |  | Информационных технологий и управления | | |
| Кафедра | |  | Систем автоматизированного  проектирования и управления | | |
| Учебная дисциплина | |  | Основы разработки автоматизированных информационных систем | | |
| Курс | 1 | | | Группа | 499м |

Лабораторная работа 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Тема: | Лабораторная работа 4 по курсу «Современные технологии разработки ПО» | | | | |
| Студент | |  |  |  | Плеханов А.А. |
| Преподаватели | |  |  |  | Корниенко И.Г., Федин К.А. |

Санкт-Петербург

2019

**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

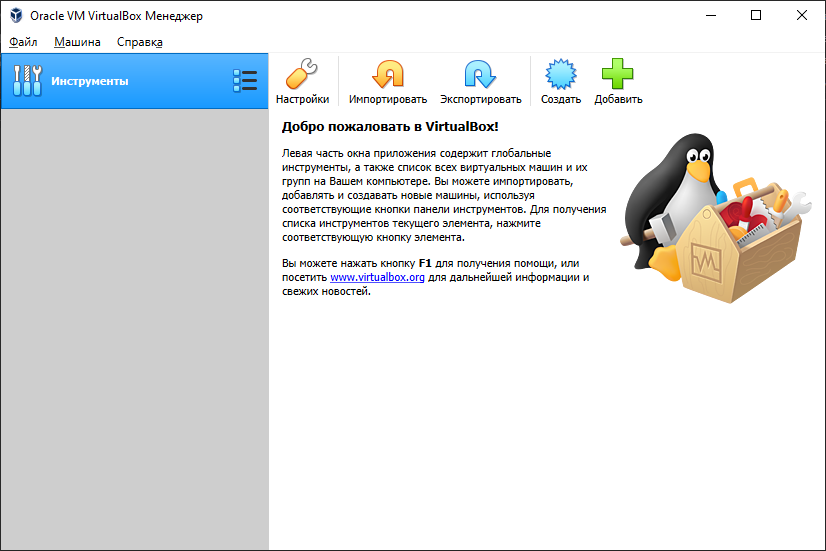


Рисунок 1 – Главное окно программы Oracle VM VirtualBox

Создание виртуальной машины

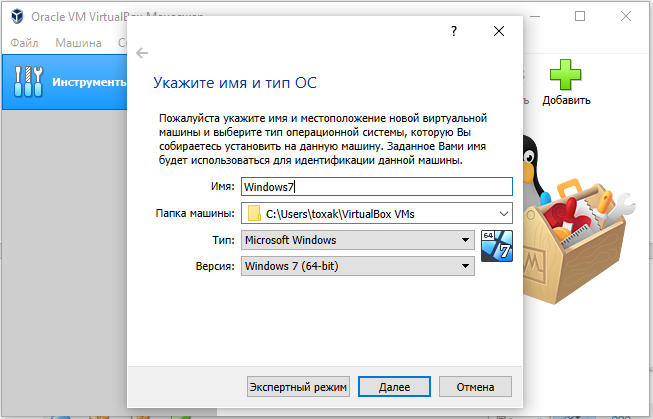


Рисунок 2 – Выбор ОС

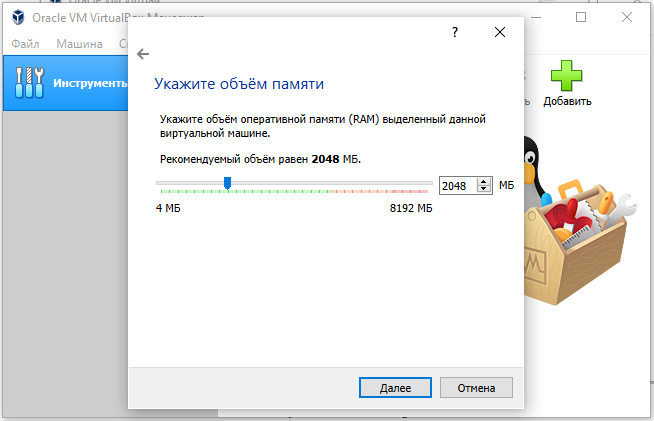


Рисунок 3 – Выбор объема памяти

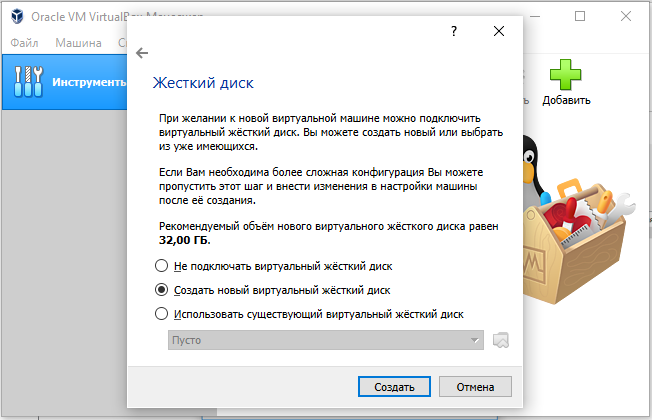
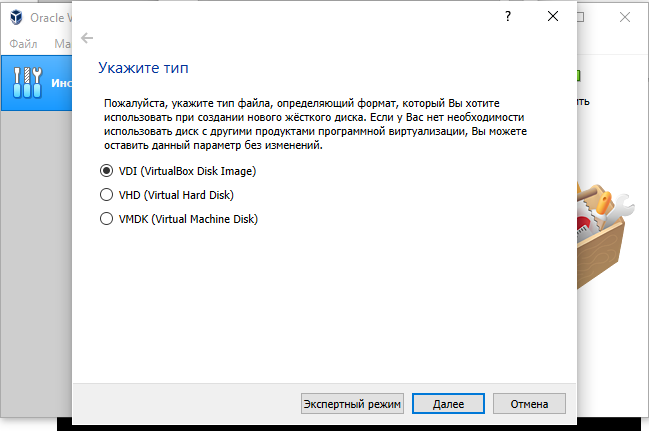


Рисунок 4 – Подключение ЖД

Рисунок 5 – Тип ЖД

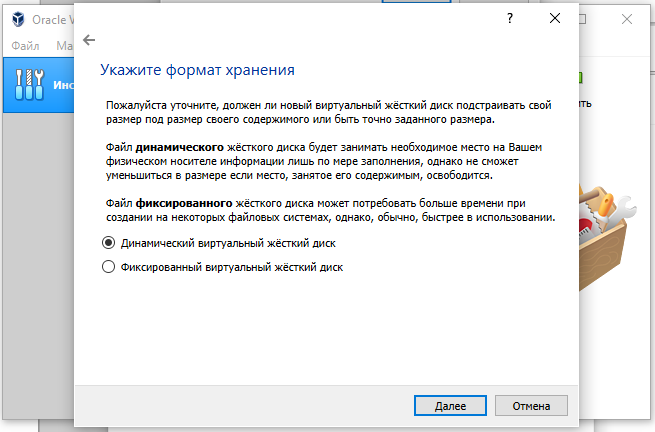


Рисунок 6 – Формат хранения

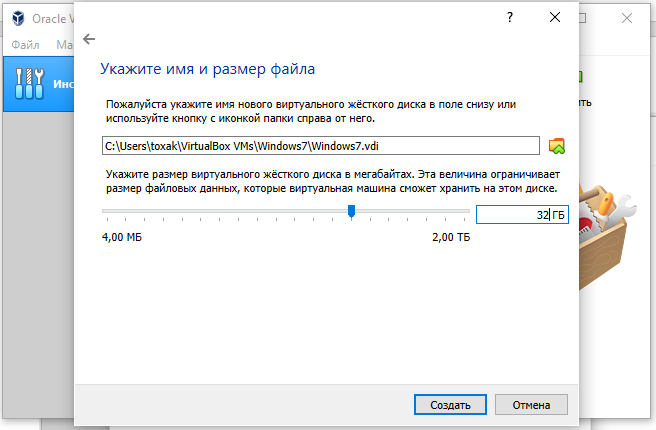


Рисунок 7 – Имя и размер ЖД

Инсталляция гостевой операционной системы

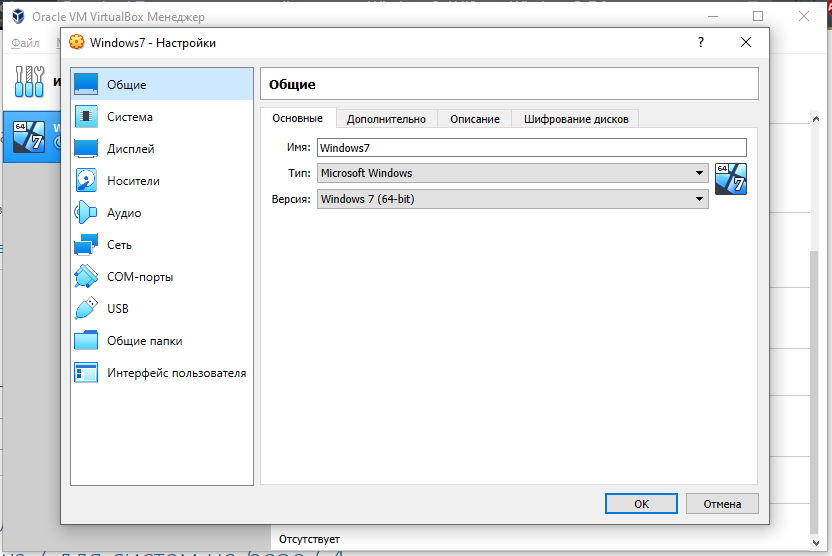


Рисунок 8 – Настройки виртуальной машины

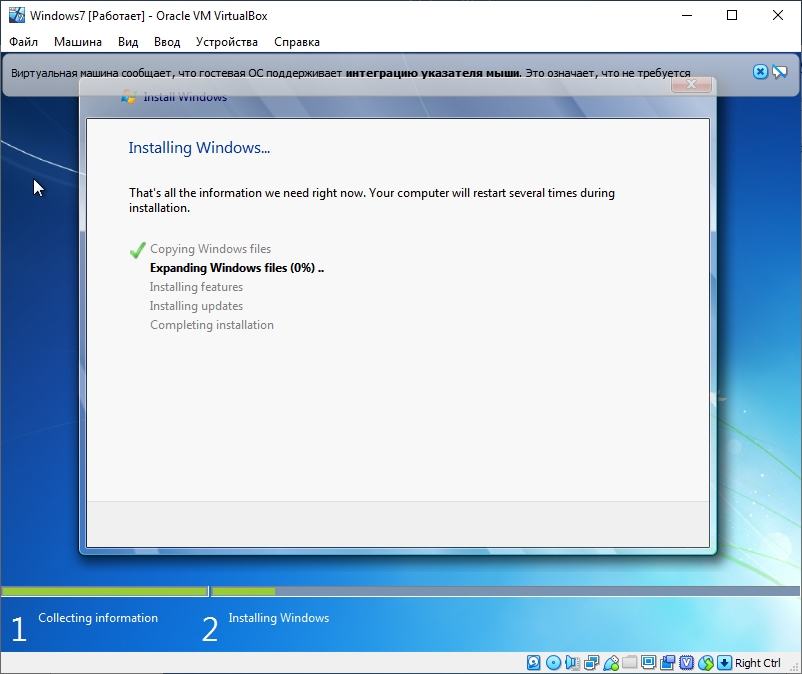


Рисунок 9 – Установка ОС на виртуальную машину

Работа на виртуальной машине

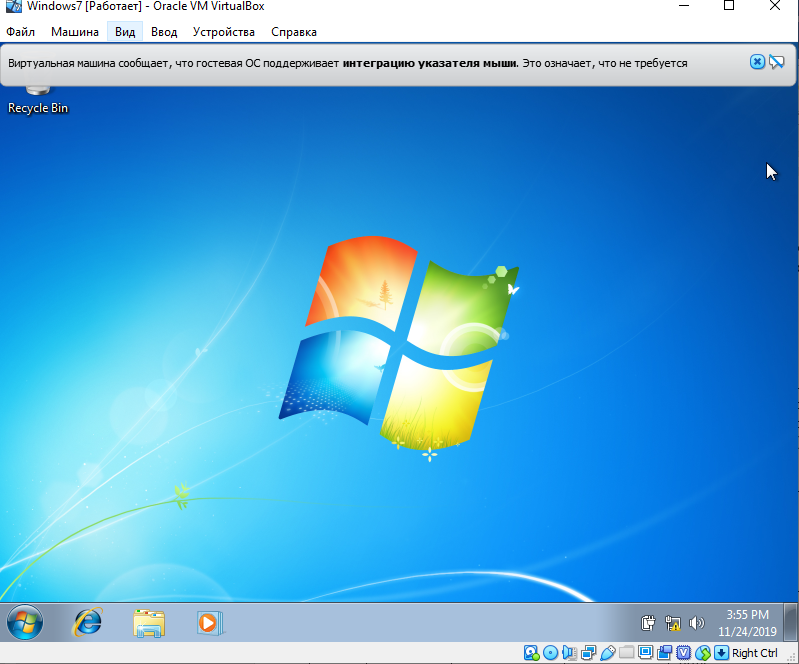


Рисунок 10 – Работа на виртуальной машине

**Листинг программы для связи с БД**

На базе предыдущих лабораторных работы была разработана программа для работы с базой данных. Ниже приведен листинг этой программы на примере модуля работы с базой экструдеров.

public class Extruder

{

[Key, DatabaseGenerated(DatabaseGeneratedOption.Identity)]

public Guid Id { get; set; }

public string Name { get; set; }

[Display(Name = "Allowed min width")]

public decimal WidthMin { get; set; }

[Display(Name = "Allowed max width")]

public decimal WidthMax { get; set; }

[Display(Name = "Allowed min thickness")]

public decimal ThicknessMin { get; set; }

[Display(Name = "Allowed max thickness")]

public decimal ThicknessMax { get; set; }

[Display(Name = "Allowed min production speed")]

public decimal ProductionSpeedMin { get; set; }

[Display(Name = "Allowed max production speed")]

public decimal ProductionSpeedMax { get; set; }

public decimal DiameterMin { get; set; }

public decimal DiameterMax { get; set; }

public decimal WeightMin { get; set; }

public decimal WeightMax { get; set; }

public decimal LengthMin { get; set; }

public decimal LengthMax { get; set; }

[Display(Name = "Time to adjust width")]

public int WidthAdjustmentTime { get; set; }

[Display(Name = "Time to change thickness")]

public int ChangeOfThicknessTime { get; set; }

[Display(Name = "Startup delay")]

public int StartupDelay { get; set; }

[Display(Name = "Machine hour cost")]

public decimal MachineHourCost { get; set; }

public decimal WidthAdjustmentConsumption { get; set; }

public decimal ChangeOfThicknessConsumption { get; set; }

public List<ExtruderCalibrationChange> ExtruderCalibrationChange { get; set; }

public List<ExtruderCoolingLipChange> ExtruderCoolingLipChange { get; set; }

public List<ExtruderNozzleChange> ExtruderNozzleChange { get; set; }

public List<ExtruderRecipeChange> ExtruderRecipeChange { get; set; }

[NotMapped]

public bool Selected { get; set; }

public Extruder()

{

ExtruderCalibrationChange = new List<ExtruderCalibrationChange>();

ExtruderCoolingLipChange = new List<ExtruderCoolingLipChange>();

ExtruderNozzleChange = new List<ExtruderNozzleChange>();

ExtruderRecipeChange = new List<ExtruderRecipeChange>();

}

}

[AuthorizeRoles]

public class ExtrudersController : Controller

{

private OptelContext db = new OptelContext();

// GET: Extruders

public async Task<ActionResult> Index(int? page)

{

var pageNumber = page ?? 1;

var pageContent = await db.Extruders.OrderBy(i => i.Name).ToPagedListAsync(pageNumber, Convert.ToInt32(WebConfigurationManager.AppSettings["ElementsPerIndexPage"]));

ViewBag.PageContent = pageContent;

return View();

}

// GET: Extruders/Details/5

public async Task<ActionResult> Details(Guid? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Extruder extruder = await db.Extruders.FindAsync(id);

if (extruder == null)

{

return HttpNotFound();

}

return View(extruder);

}

[AuthorizeRoles(Utils.User.Roles.Admin)]

// GET: Extruders/Create

public ActionResult Create()

{

return View();

}

// POST: Extruders/Create

// Чтобы защититься от атак чрезмерной передачи данных, включите определенные свойства, для которых следует установить привязку. Дополнительные

// сведения см. в статье https://go.microsoft.com/fwlink/?LinkId=317598.

[HttpPost]

[ValidateAntiForgeryToken]

public async Task<ActionResult> Create([Bind(Include = "Id,Name,WidthMin,WidthMax,ThicknessMin,ThicknessMax,ProductionSpeedMin,ProductionSpeedMax,WidthAdjustmentTime,ChangeOfThicknessTime,StartupDelay,MachineHourCost")] Extruder extruder)

{

if (extruder.ThicknessMin >= extruder.ThicknessMax)

{

ModelState.AddModelError("ThicknessMin", "Allowed min thickness must be smaller than allowed max thickness.");

}

if (extruder.ProductionSpeedMin >= extruder.ProductionSpeedMax)

{

ModelState.AddModelError("ProductionSpeedMin", "Allowed min production speed must be smaller than allowed max production speed.");

}

if (extruder.WidthMin >= extruder.WidthMax)

{

ModelState.AddModelError("WidthMin", "Allowed min width must be smaller than allowed max width.");

}

if (ModelState.IsValid)

{

extruder.Id = Guid.NewGuid();

db.Extruders.Add(extruder);

await db.SaveChangesAsync();

return RedirectToAction("Index");

}

return View(extruder);

}

[AuthorizeRoles(Utils.User.Roles.Admin)]

// GET: Extruders/Edit/5

public async Task<ActionResult> Edit(Guid? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Extruder extruder = await db.Extruders.FindAsync(id);

if (extruder == null)

{

return HttpNotFound();

}

return View(extruder);

}

[AuthorizeRoles(Utils.User.Roles.Admin)]

// POST: Extruders/Edit/5

// Чтобы защититься от атак чрезмерной передачи данных, включите определенные свойства, для которых следует установить привязку. Дополнительные

// сведения см. в статье https://go.microsoft.com/fwlink/?LinkId=317598.

[HttpPost]

[ValidateAntiForgeryToken]

public async Task<ActionResult> Edit([Bind(Include = "Id,Name,WidthMin,WidthMax,ThicknessMin,ThicknessMax,ProductionSpeedMin,ProductionSpeedMax,DiameterMin,DiameterMax,WeightMin,WeightMax,LenghtMin,LenghtMax,WidthAdjustmentTime,ChangeOfThicknessTime,StartupDelay,MachineHourCost,WidthAdjustmentConsumption,ChangeOfThicknessTimeConsumption")] Extruder extruder)

{

if (extruder.ThicknessMin >= extruder.ThicknessMax)

{

ModelState.AddModelError("ThicknessMin", "Allowed min thickness must be smaller than allowed max thickness.");

}

if (extruder.ProductionSpeedMin >= extruder.ProductionSpeedMax)

{

ModelState.AddModelError("ProductionSpeedMin", "Allowed min production speed must be smaller than allowed max production speed.");

}

if (extruder.WidthMin >= extruder.WidthMax)

{

ModelState.AddModelError("WidthMin", "Allowed min width must be smaller than allowed max width.");

}

if (ModelState.IsValid)

{

db.Entry(extruder).State = EntityState.Modified;

await db.SaveChangesAsync();

return RedirectToAction("Index");

}

return View(extruder);

}

[AuthorizeRoles(Utils.User.Roles.Admin)]

// GET: Extruders/Delete/5

public async Task<ActionResult> Delete(Guid? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Extruder extruder = await db.Extruders.FindAsync(id);

if (extruder == null)

{

return HttpNotFound();

}

return View(extruder);

}

[AuthorizeRoles(Utils.User.Roles.Admin)]

// POST: Extruders/Delete/5

[HttpPost, ActionName("Delete")]

[ValidateAntiForgeryToken]

public async Task<ActionResult> DeleteConfirmed(Guid id)

{

Extruder extruder = await db.Extruders.FindAsync(id);

db.Extruders.Remove(extruder);

await db.SaveChangesAsync();

return RedirectToAction("Index");

}

protected override void Dispose(bool disposing)

{

if (disposing)

{

db.Dispose();

}

base.Dispose(disposing);

}

}