Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Уфимский университет науки и технологии»

Отчет по лабораторным работам №1-8

По дисциплине «Информационные системы и технологии»

Выполнили: студенты группы ИСТ-318Б

Акатьев Б.А

Попов А.А.

Проверила: доктор технических наук

Христодуло О.И.

Уфа 2024

**Содержание**

[Лабораторная работа №1. «ПРОГРАММИРОВАНИЯ JAVA. ОСНОВНЫЕ ОПЕРАТОРЫ РАБОТЫ С ДАННЫМИ» 3](#_Toc184581749)

[Задание №1 - Сумма чисел 3](#_Toc184581750)

[Задание №7 – Таблица температур 5](#_Toc184581751)

[Задание №10 – Количество делителей 6](#_Toc184581752)

[Задание №2 – Сумма функций 7](#_Toc184581753)

[Вывод 9](#_Toc184581754)

[Лабораторная работа №2. «СЕРВЕРНЫЙ ЯЗЫК НАПИСАНИЯ СЦЕНАРИЕВ PHP» 10](#_Toc184581755)

[Задание №1 – Поиск по документу 10](#_Toc184581756)

[Вывод 13](#_Toc184581757)

[Лабораторная работа №3. «РАБОТА С ГРАФИКОЙ C#» 14](#_Toc184581758)

[Задание 1 - Спираль 14](#_Toc184581759)

[Задание 2 – Светофор 20](#_Toc184581760)

[Задание 3 – Просмотр графических файлов 27](#_Toc184581761)

[Вывод 31](#_Toc184581762)

[Лабораторная работа №4. «ГРАФИЧЕСКИЕ ПРИМИТИВЫ В C#» 32](#_Toc184581763)

[Задание 1 - Построение графических примитивов 32](#_Toc184581764)

[Лабораторная работа №6. Установка и первичная настройка Windows Server как контроллера домена 35](#_Toc184581765)

[Лабораторная работа №7. Администрирование компьютеров и файлового сервера в домене Windows 43](#_Toc184581766)

[**Лабораторная работа №8. Мониторинг файлового сервера. Автономные файлы. Службы DNS и DHCP.** 50](#_Toc184581767)

Лабораторная работа №1. «ПРОГРАММИРОВАНИЯ JAVA. ОСНОВНЫЕ ОПЕРАТОРЫ РАБОТЫ С ДАННЫМИ»

**Цель:**

Изучить основные операторы работы с данными в языке  
программирования Java.

**Ход выполнения лабораторной работы:**

Задание №1 - Сумма чисел

Определить сумму чисел от 3 до 99 кратных числу 3.

Исходный код программы:

import java.util.Scanner;

public class Main

{

public static void main(String[] args) {

Scanner s = new Scanner(System.in);

System.out.print("Задание №1 \n");

int sum = 0;

for(int y = 3; y < 100; y =y+ 3)

{

sum = sum + y;

}

System.out.print("Сумма от 3 до 99 равна: "+ sum +" \n");

}

}

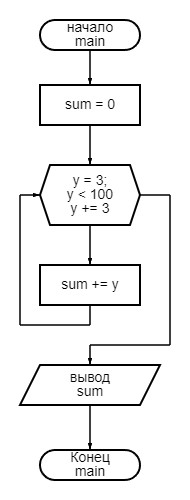


Рисунок 1. Блок-схема программы для задания №1

Задание №7 – Таблица температур

Напечатать таблицы температур по Цельсию от 0 до 100 градусов с дискретностью в один градус и их эквивалентов по шкале Кельвина   
  
Исходный код программы:

**public** **static** **void** printTemperatureTable() {

System.***out***.printf("%-15s %-15s\n", "Цельсий (°C)", "Кельвин (K)");

**for** (**int** celsius = 0; celsius <= 100; celsius++) {

**double** kelvin = celsius + 273.15;

System.***out***.printf("%-15d %-15.2f\n", celsius, kelvin);

}

}

Изображение выглядит как диаграмма, зарисовка, Шрифт, дизайн

Автоматически созданное описание

Рисунок 2. Блок-схема программы для задания №7

Задание №10 – Количество делителей

Задано число P. Определить его количество делителей.

Исходный код программы:

**public** **static** **void** deliteli() {

Scanner s = **new** Scanner(System.***in***);

System.***out***.print("Задание №10 \n Введите число P, для которого требуется определить кол-во делителей. \n");

**int** P = s.nextInt();

**int** counter = 1;

**for**(**int** i = 1;i < P;i += 1)

{

**if** (P % i == 0)

{

counter += 1;

}

}

}

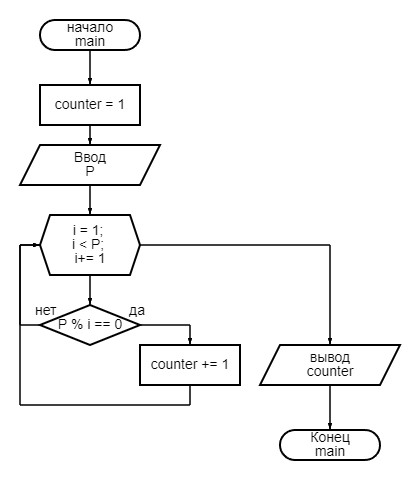
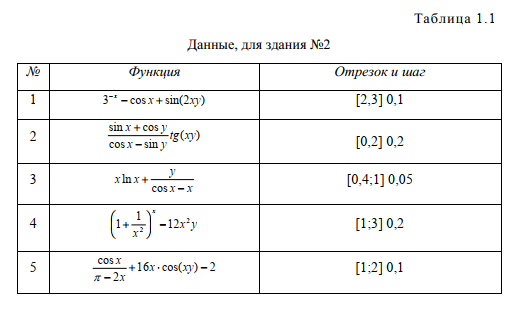


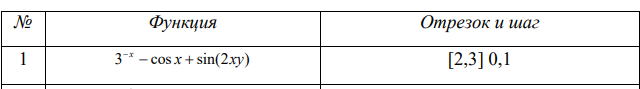
Рисунок 3. Блок-схема программы для задания №10

Задание №2 – Сумма функций

Посчитать сумму по приведенным ниже данным  
(y=const) (см. табл. 1.1):



Для выполнения задания №2 был выбран вариант 1:



Исходный код программы:

**package** akatev\_ist1lab;

**import** java.util.Scanner;

**public** **class** zad1 {

**public** **static** **void** main(String[] args) {

Scanner s = **new** Scanner(System.***in***);

// Ввод значения переменной y (константа)

System.***out***.print("Введите значение переменной y: ");

**float** y = s.nextFloat();

// Задаем начальное и конечное значения x, а также шаг

**float** x\_min = 2.0f;

**float** x\_max = 3.0f;

**double** step = 0.1;

**double** result;

**double** sum = 0;

// Цикл по x с шагом 0.1

**for** (**double** x = x\_min; x <= x\_max; x += step) {

// Вычисляем значение функции f(x, y)

result = 3 \* Math.*cos*(x) \* Math.*sin*(2 \* x \* y);

sum += result; // Добавляем результат к сумме

}

// Выводим итоговую сумму

System.***out***.println("Сумма функции: " + sum);

}

}

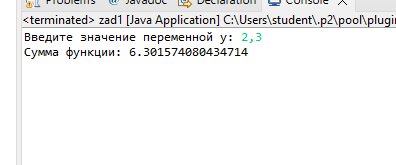


Рисунок 4*.* Результат работы программы для задания №2

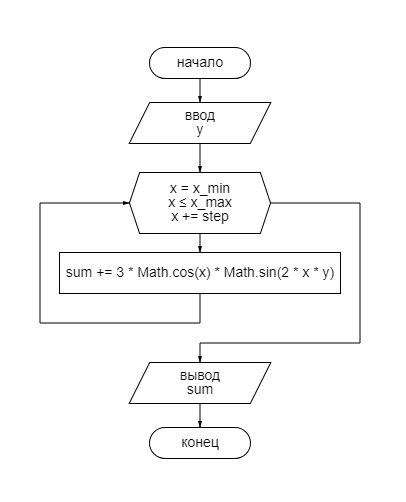


Рисунок 5. Блок-схема программы для задания №2

Вывод

В ходе выполнения Лабораторной работы №1 были изучены основы языка программирования Java, а так же его основные операторы работы с данными.

Лабораторная работа №2. «СЕРВЕРНЫЙ ЯЗЫК НАПИСАНИЯ СЦЕНАРИЕВ PHP»

**Цель:**

Изучить основы работы с PHPDesigner.

**Ход выполнения лабораторной работы:**

Задание №1 – Поиск по документу

Создать сценарий, позволяющий производить поиск по документу.

Пишем функцию для поиска по документу, заходим в документ, инициализируем функцию.

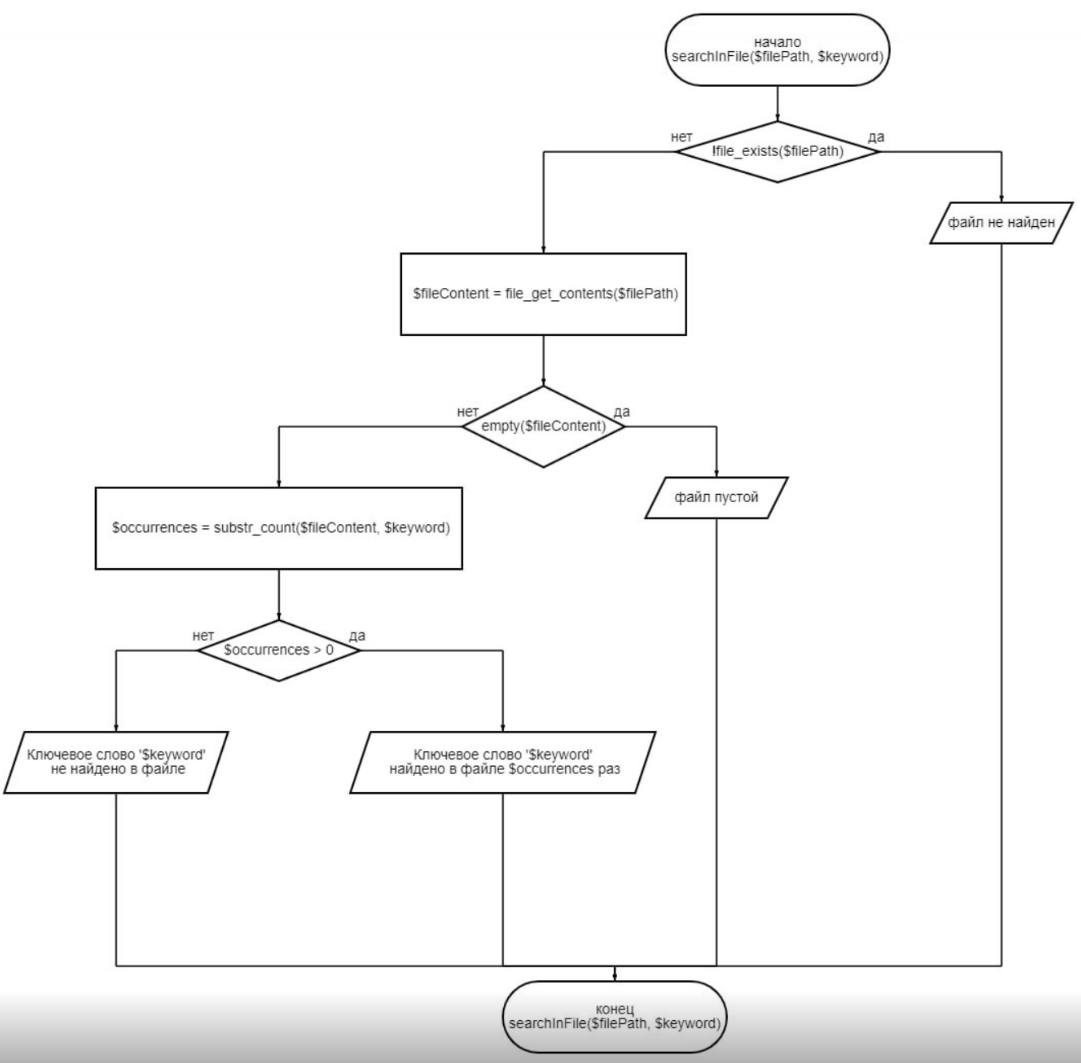
Блок-схема функции поиска:

Рисунок - Блок-схема функции поиска

Блок-схема основной функции:

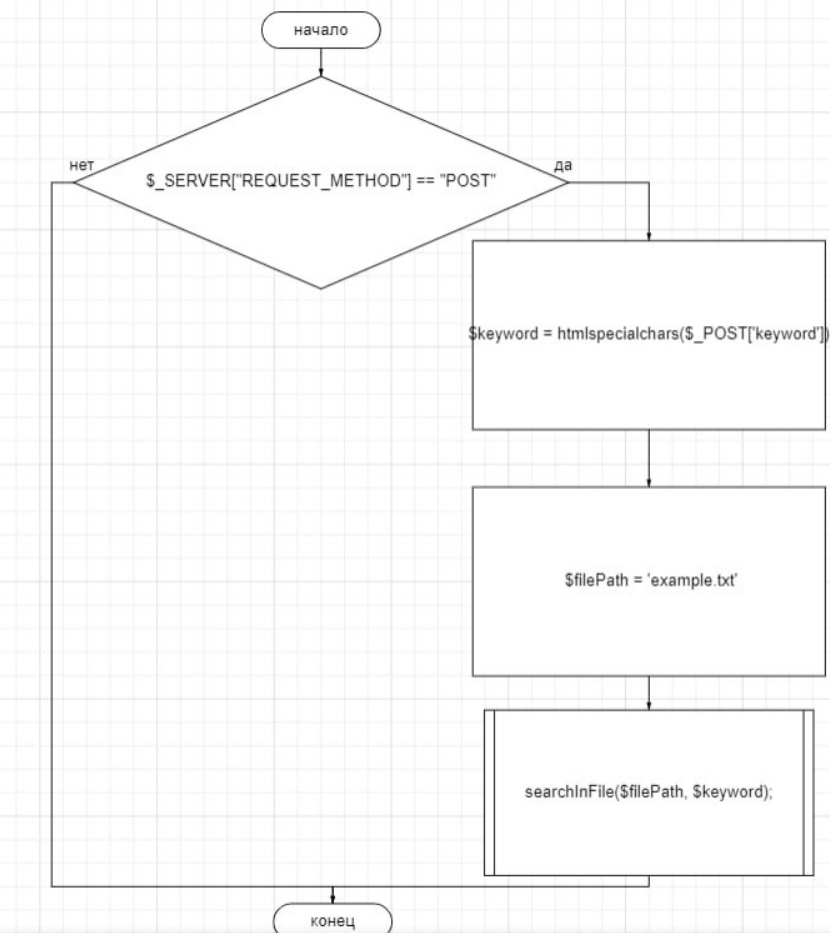


Рисунок 7 - Блок-схема основной функции

Исходный код программы:

<?php

function searchInFile($filePath, $keyword) {

if (!file\_exists($filePath)) {

return "Файл не найден.";

}

$fileContent = file\_get\_contents($filePath);

if (empty($fileContent)) {

return "Файл пустой.";

}

// Ищем ключевое слово в содержимом файла

$occurrences = substr\_count($fileContent, $keyword);

if ($occurrences > 0) {

return "Ключевое слово '$keyword' найдено в файле $occurrences раз";

} else {

return "Ключевое слово '$keyword' не найдено в файле";

}

}

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

$keyword = htmlspecialchars($\_POST['keyword']);

$filePath = 'example.txt';

echo searchInFile($filePath, $keyword);

}

?>

Результат:

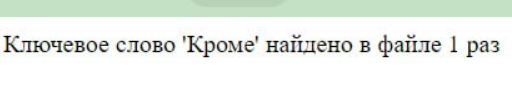


Рисунок 8 – Результат выполнения программы

Вывод

В ходе выполнения Лабораторной работы №2 были изучены основы работы с PHPDesigner, а так же создана собственная программа с функцией для поиска вхождения слов по документу.

Лабораторная работа №3. «РАБОТА С ГРАФИКОЙ C#»

**Цель:**

Целью работы является изучение возможностей разработки графических приложений на языке C#.

Задание 1 - Спираль

Создать программу, позволяющий выводить на экран спираль Архимеда с заданными на элементе TrackBar показателями число витков и масштаб

**Решение:** с помощью элементов Windows Form и языка программирования C# была создана программа, выводящая на экран спираль Архимеда.

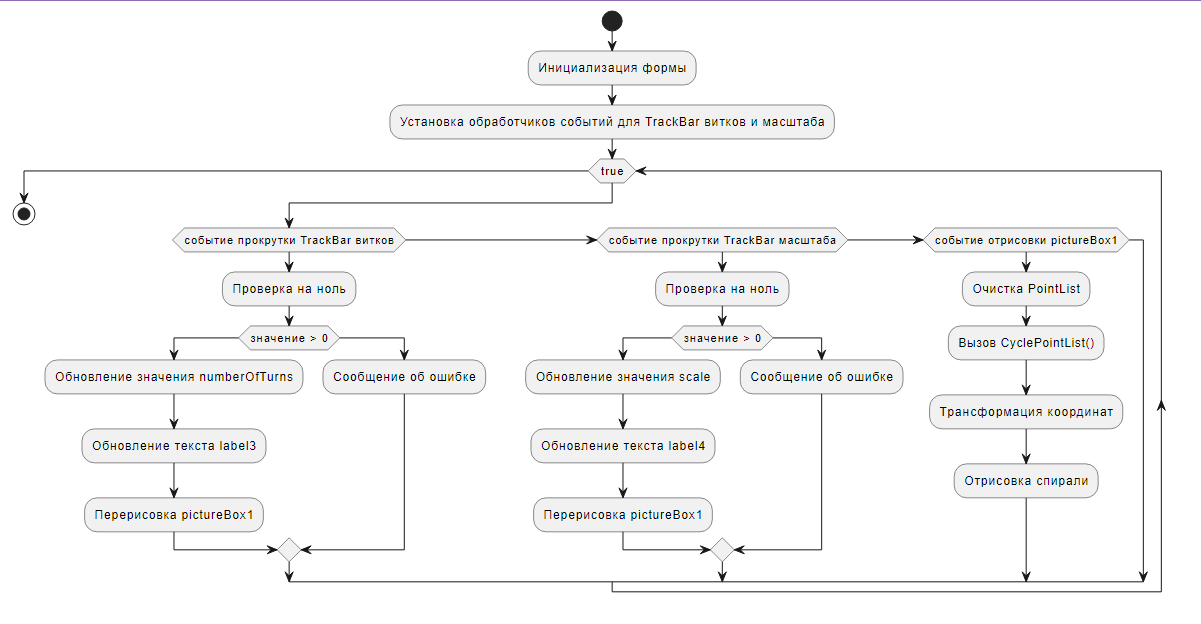
****

Рисунок 9 – Блок схема программы

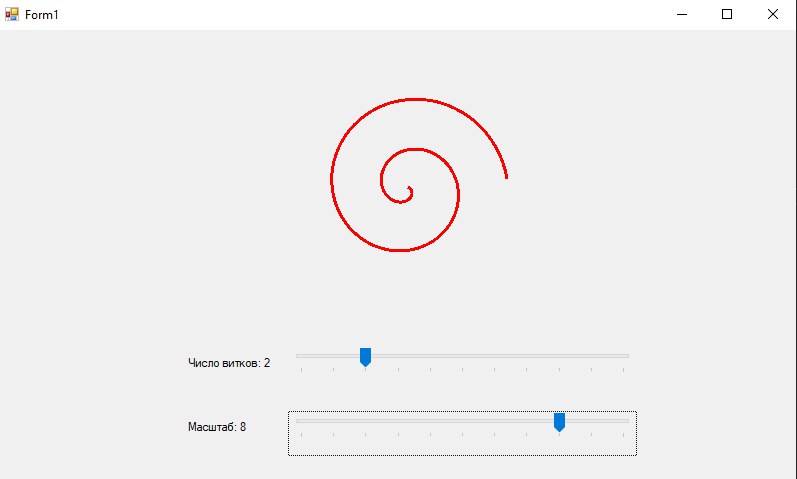
****

Рисунок 10 – Пример выполнения программы

Исходный код программы:

﻿using System;

using System.Collections.Generic;

using System.Drawing;

using System.Windows.Forms;

namespace svetofor

{

public partial class Form1 : Form

{

private List<PointF> PointList;

private Pen redPen = new Pen(Color.Red, 3);

private int numberOfTurns = 1; // Начальное значение 1 виток

private float scale = 10; // Начальное значение масштаба

public Form1()

{

InitializeComponent();

PointList = new List<PointF>();

pictureBox1.Paint += PictureBox1\_Paint;

trackBar1.Scroll += TrackBar1\_Scroll; // Подписка на событие прокрутки для TrackBar витков

trackBar2.Scroll += TrackBar2\_Scroll; // Подписка на событие прокрутки для TrackBar масштаба

}

private void PictureBox1\_Paint(object sender, PaintEventArgs e)

{

DrawSpiral(e.Graphics);

}

private void DrawSpiral(Graphics g)

{

PointList.Clear();

CyclePointList();

g.TranslateTransform(pictureBox1.ClientSize.Width / 2f, pictureBox1.ClientSize.Height / 2f);

g.DrawCurve(redPen, PointList.ToArray());

}

private void CyclePointList()

{

// Проверка на количество витков и масштаб

if (numberOfTurns <= 0 || scale <= 0)

{

return; // Если число витков 0 или меньше, или масштаб 0 или меньше, выходим из метода

}

for (float angle = 0; angle < numberOfTurns \* 360; angle += 5)

{

float radian = angle \* (float)Math.PI / 180;

float x = (float)(radian \* Math.Cos(radian) \* scale);

float y = (float)(radian \* Math.Sin(radian) \* scale);

PointList.Add(new PointF(x, y));

}

}

private void TrackBar1\_Scroll(object sender, EventArgs e)

{

if (trackBar1.Value == 0)

{

MessageBox.Show("Число витков не может быть равно нулю.");

return; // Игнорируем изменение до нуля

}

numberOfTurns = trackBar1.Value;

label3.Text = $"Число витков: {numberOfTurns}";

pictureBox1.Invalidate(); // Перерисовываем PictureBox

}

private void TrackBar2\_Scroll(object sender, EventArgs e)

{

if (trackBar2.Value == 0)

{

MessageBox.Show("Масштаб не может быть равен нулю.");

return; // Игнорируем изменение до нуля

}

scale = trackBar2.Value;

label4.Text = $"Масштаб: {scale}";

pictureBox1.Invalidate();

}

private void Form1\_Load(object sender, EventArgs e)

{

trackBar1.Minimum = 1; // Минимальное значение для числа витков

trackBar1.Maximum = 10;

trackBar1.Value = numberOfTurns; // Установка начального значения на 1

trackBar2.Minimum = 1; // Минимальное значение для масштаба

trackBar2.Maximum = 50;

trackBar2.Value = (int)scale;

label3.Text = $"Число витков: {numberOfTurns}";

label4.Text = $"Масштаб: {scale}";

}

private void label1\_Click(object sender, EventArgs e)

{

MessageBox.Show("Label clicked!"); // Пример действия при клике на label

}

}

}

Задание 2 – Светофор

Создать программу, позволяющий выводить на экран светофор с переключаемыми его сигналами по определенным кнопкам

**Решение:** с помощью элементов Windows Form и языка программирования C# была создана программа, при запуске которой мы можем переключать цвета, изображенного на ней светофора

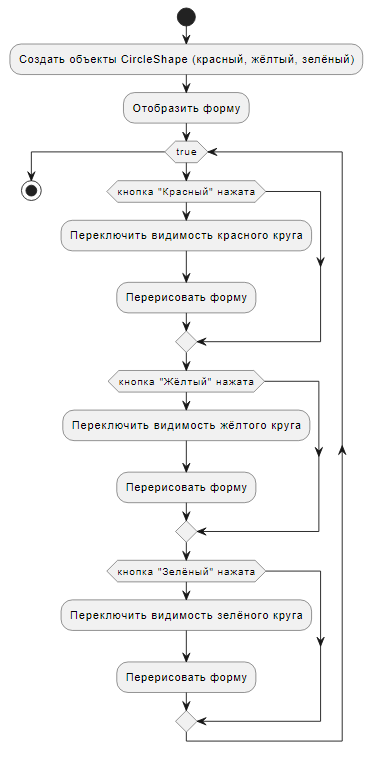


Рисунок 11 – Блок схема программы

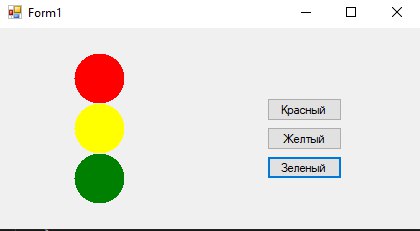


Рисунок 12 – Результат работы программы

Исходный код программы:

﻿using System;

using System.Drawing;

using System.Windows.Forms;

namespace Svetafor

{

public partial class Form1 : Form

{

private CircleShape redCircle;

private CircleShape yellowCircle;

private CircleShape greenCircle;

public Form1()

{

InitializeComponent();

InitializeTrafficLight();

this.Paint += new PaintEventHandler(Form1\_Paint);

}

private void InitializeTrafficLight()

{

redCircle = new CircleShape(this, 100, 50, Color.Red);

yellowCircle = new CircleShape(this, 100, 100, Color.Yellow);

greenCircle = new CircleShape(this, 100, 150, Color.Green);

}

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

{

redCircle.Toggle();

this.Invalidate();

}

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

{

yellowCircle.Toggle();

this.Invalidate();

}

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

{

greenCircle.Toggle();

this.Invalidate();

}

private void Form1\_Paint(object sender, PaintEventArgs e)

{

redCircle.Draw(e.Graphics);

yellowCircle.Draw(e.Graphics);

greenCircle.Draw(e.Graphics);

}

private class CircleShape

{

private Form form;

private int x;

private int y;

private Color color;

private bool visible;

public CircleShape(Form form, int x, int y, Color color)

{

this.form = form;

this.x = x;

this.y = y;

this.color = color;

this.visible = false;

}

public void Draw(Graphics g)

{

if (visible)

{

using (SolidBrush brush = new SolidBrush(color))

{

g.FillEllipse(brush, x - 25, y - 25, 50, 50);

}

}

}

public void Show()

{

visible = true;

form.Invalidate();

}

public void Hide()

{

visible = false;

form.Invalidate();

}

public void Toggle()

{

visible = !visible; // Переключаем видимость

form.Invalidate();

}

}

}

}

Задание 3 – Просмотр графических файлов

Решение: с помощью wpf и элементов label, Picturebox, реализована внешняя часть приложения.

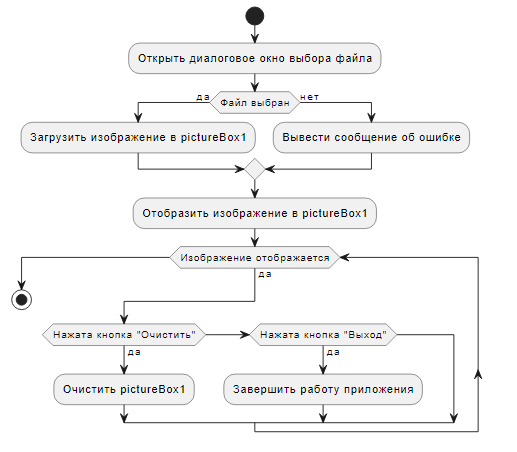
****

Рисунок 13 – Блок схема программы

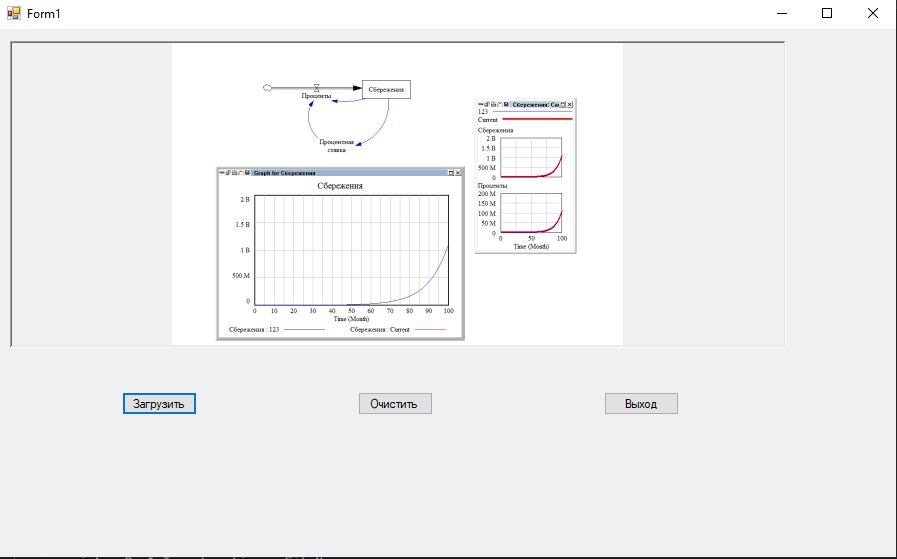
****

Рисунок 14 – Результат выполнения программы

Исходный код программы:

﻿using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace loader

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

OpenFileDialog openFile = new OpenFileDialog();

openFile.ShowDialog();

string FileName = openFile.FileName;

pictureBox1.ImageLocation = FileName;

}

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

{

pictureBox1.Image = null;

}

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

{

Application.Exit();

}

}

}

Вывод

В ходе выполнения Лабораторной работы №3 были изучены основы работы с WindowsForm, и созданы 3 программы:   
1 – программа для отображения спирали Архимеда

2 – программа для отображения работы светофора

3 – программа для выбора и отображения графического файла

Лабораторная работа №4. «ГРАФИЧЕСКИЕ ПРИМИТИВЫ В C#»

**Цель:**

Целью работы является изучение возможностей разработки графических приложений, строящих графические примитивы.

Задание 1 - Построение графических примитивов

Создать программу, позволяющий выводить на экран элемент PictureBox графические примитивы в зависимости от положения элемента RadioButton. Границы координат фигур задаются пользователем с помощью щелчка левой кнопки мыши по PictureBox.

**#########**

Рисунок 13 – Блок схема программы

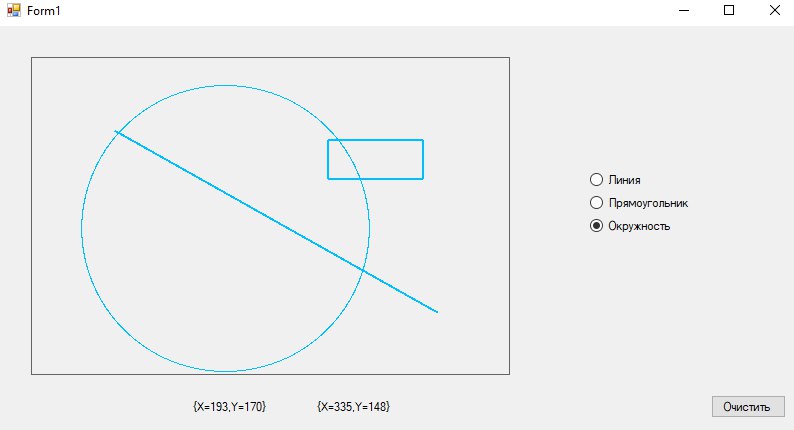
****

Рисунок 14 – Результат выполнения программы

Исходный код программы:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Reflection.Emit;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using static System.Windows.Forms.VisualStyles.VisualStyleElement.Button;

namespace WinFormsApp1

{

public partial class Form1 : Form

{

public bool \_click = false;

public Point location1;

public Point location2;

public Graphics g;

public Form1()

{

InitializeComponent();

}

private void radioButton1\_CheckedChanged(object sender, EventArgs e)

{

}

private void radioButton2\_CheckedChanged(object sender, EventArgs e)

{

}

private void radioButton3\_CheckedChanged(object sender, EventArgs e)

{

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void label2\_Click(object sender, EventArgs e)

{

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

pictureBox1.MouseClick += OnPictureBoxClicked;

}

private void OnPictureBoxClicked(object sender, MouseEventArgs args)

{

if (\_click == false)

{

\_click = true;

location1 = args.Location;

pictureBox1.MouseClick -= OnPictureBoxClicked;

label1.Text = location1.ToString();

}

else

{

\_click = false;

location2 = args.Location;

pictureBox1.MouseClick -= OnPictureBoxClicked;

label2.Text = location2.ToString();

Drawing();

}

}

private void Drawing()

{

Pen pen = new Pen(Brushes.DeepSkyBlue);

pen.Width = 2.0F;

g = Graphics.FromHwnd(pictureBox1.Handle);

if (radioButton1.Checked == true)

{

g.DrawLine(pen, location1, location2);

}

else if (radioButton2.Checked == true)

{

g.DrawLine(pen, location1.X, location1.Y, location2.X, location1.Y);

g.DrawLine(pen, location2.X, location1.Y, location2.X, location2.Y);

g.DrawLine(pen, location2.X, location2.Y, location1.X, location2.Y);

g.DrawLine(pen, location1.X, location2.Y, location1.X, location1.Y);

//DrawLine по четырём линиям

}

else if (radioButton3.Checked == true)

{

SolidBrush brush = new SolidBrush(Color.DeepSkyBlue);

int R = (int)Math.Round(Math.Sqrt(Math.Pow((location2.X - location1.X), 2) + Math.Pow((location2.Y - location1.Y), 2)), 1);

int x = 0;

int y = R;

double delta = 1 - 2 \* R;

double error = 0;

while (y >= 0)

{

g.FillRectangle(brush, new Rectangle(location1.X + x, location1.Y + y, 1, 1));

g.FillRectangle(brush, new Rectangle(location1.X + x, location1.Y - y, 1, 1));

g.FillRectangle(brush, new Rectangle(location1.X - x, location1.Y + y, 1, 1));

g.FillRectangle(brush, new Rectangle(location1.X - x, location1.Y - y, 1, 1));

error = 2 \* (delta + y) - 1;

if ((delta < 0) && (error <= 0))

{

delta += 2 \* ++x + 1;

continue;

}

if ((delta > 0) && (error > 0))

{

delta -= 2 \* --y + 1;

continue;

}

delta += 2 \* (++x - y--);

}

}

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

pictureBox1.Image = null;

}

}

}

Лабораторная работа №6. Установка и первичная настройка Windows Server как контроллера домена

**Цель работы:** Получение практических знаний по установке и настройке операционных систем на виртуальной машине VirtualBox.

**Ход работы:**

Запускаем средства виртуализации и создаем новой виртуальной машиныи устанавливаем Windows Server 2008

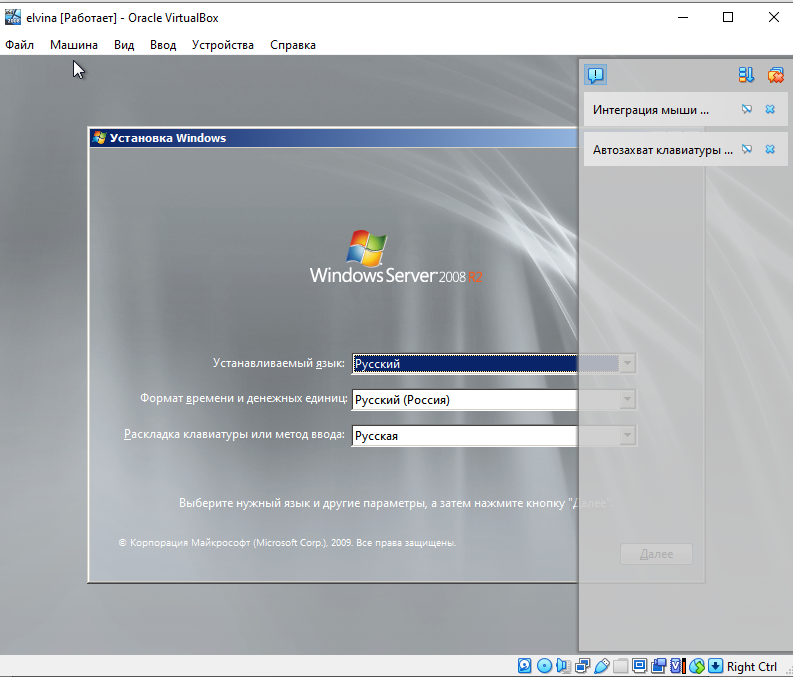


Рисунок 23. Окно установки Windows Server 2008

При установке операционной системы необходимо выбрать версию (в нашем случае Windows Server 2008 Enterprise) и диск установки.

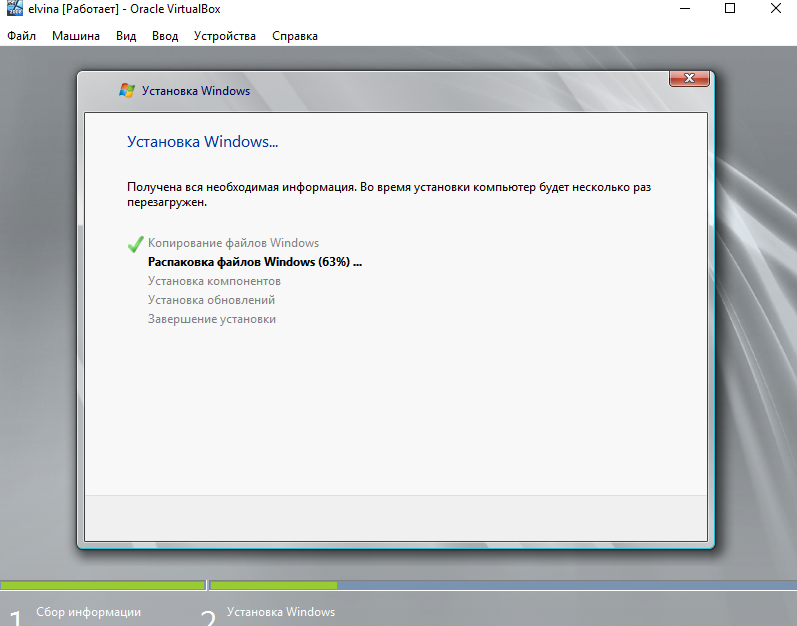


Рисунок 24 Установка ОС Windows Server 2008

Устанавливаем пароль для учетной записи «Администратор».

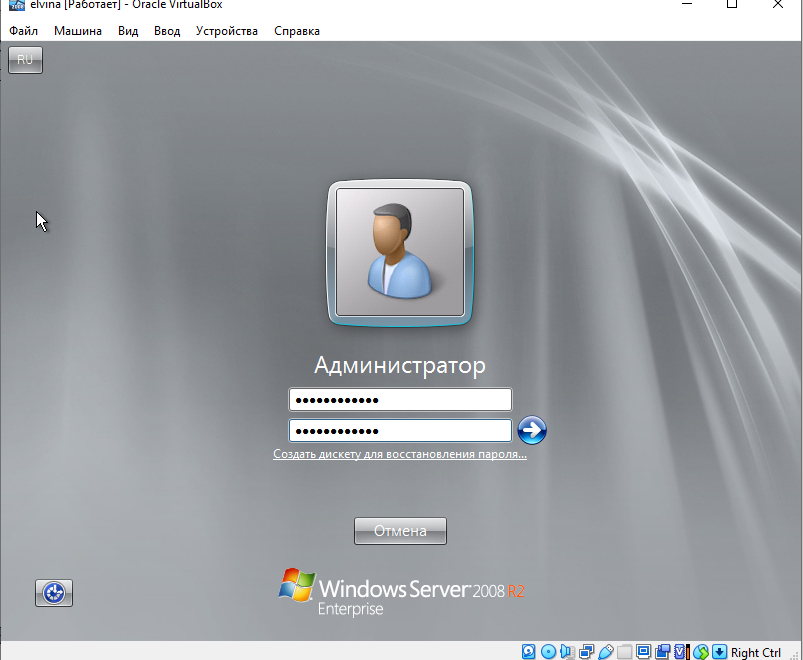


Рисунок 25 Создание пароля учётной записи «Администратор»

Меняем сгенерированное «Полное имя компьютера»

После создания Администратора необходимо установить роли и компоненты

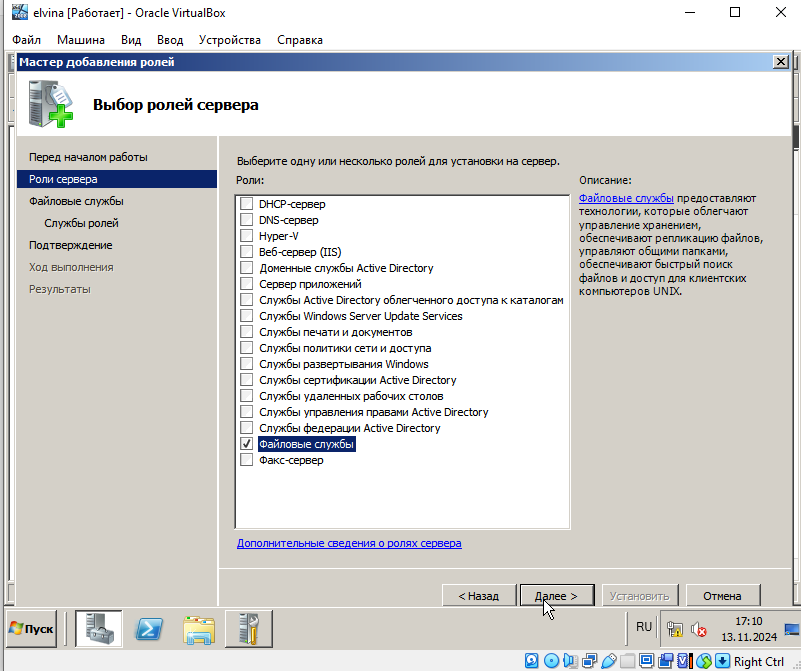


Рисунок Установка роли «Файловые службы»

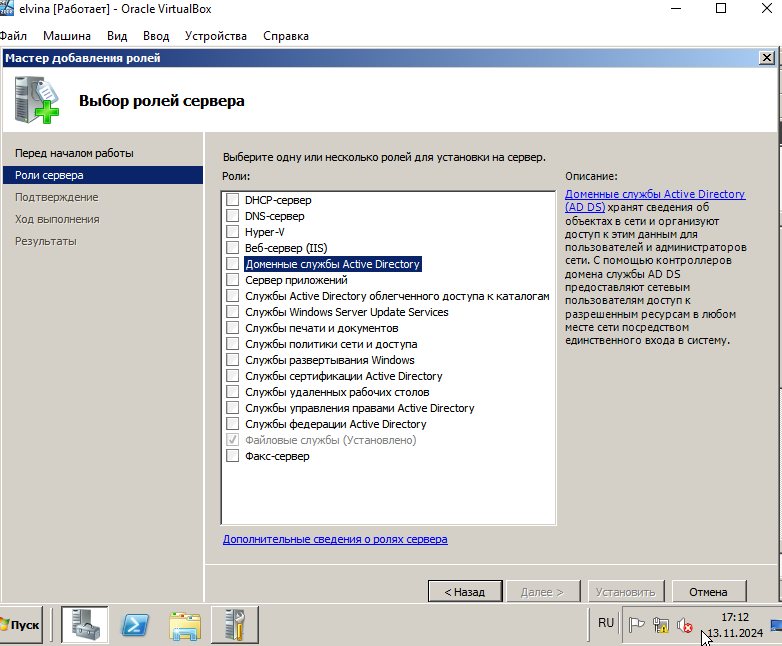


Рисунок 27 Роль «Доменные службы Active Directory»

Для установки «Доменные службы Active Directory» создаём новый домен в лесу

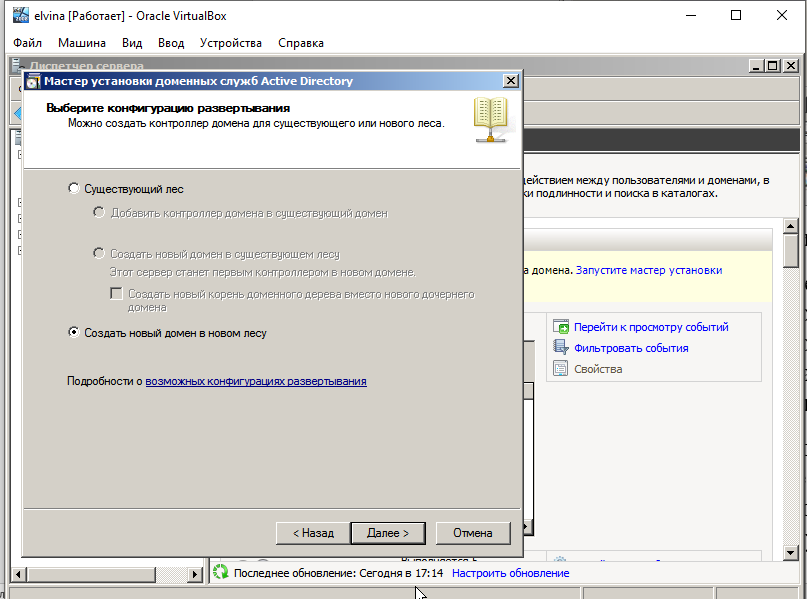


Рисунок 28 Мастер установки доменных служб Active Directory

Указываем имя корневого домена леса

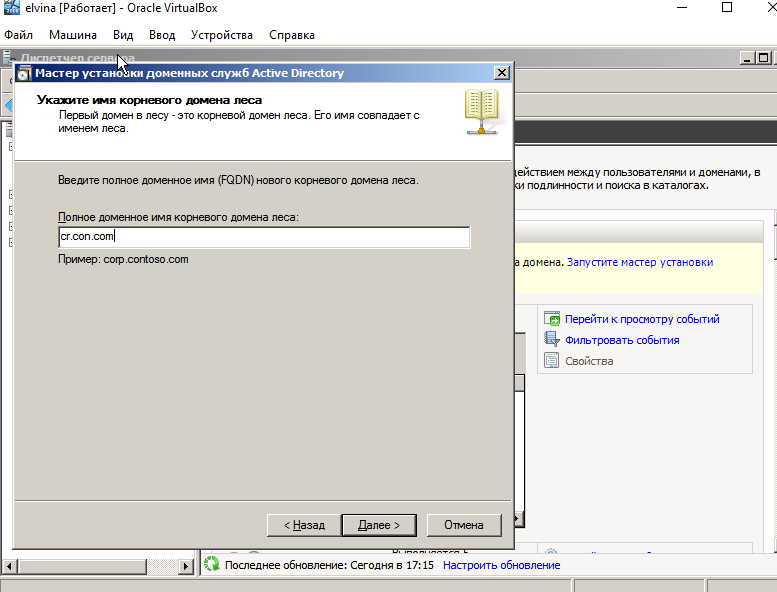


Рисунок 29 Имя корневого домена леса

Назначаем серверу статический IP-адрес

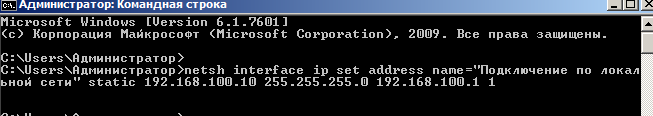


Рисунок 35 Назначенный IP-адрес

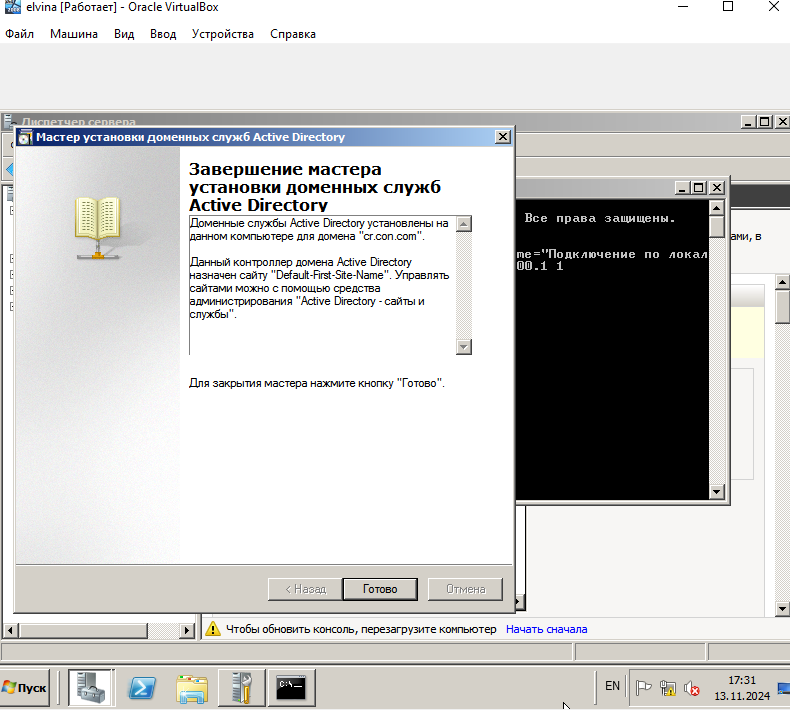


Рисунок 30 Завершение первичной настройки контроллера домена

**Вывод:** в ходе выполнения лабораторной работы №6 были получены практические знания по установке, а так же настройке операционной системы на виртуальной машине VirtualBox.

Лабораторная работа №7. Администрирование компьютеров и файлового сервера в домене Windows

**Цель работы:** Получение практических знаний об администрировании файлового сервера в сетях, управляемых ОС Windows, создании доменных учётных записей, работе с ролью файлового сервера, организации квот, управлению общими папками и автономными файлами.

**Ход работы**:

Для выполнения лабораторной работы необходимо создать вторую виртуальную машину с Windows 7, которая будет выполнять роль клиента.

Установим ОС на вторую виртуальную машину

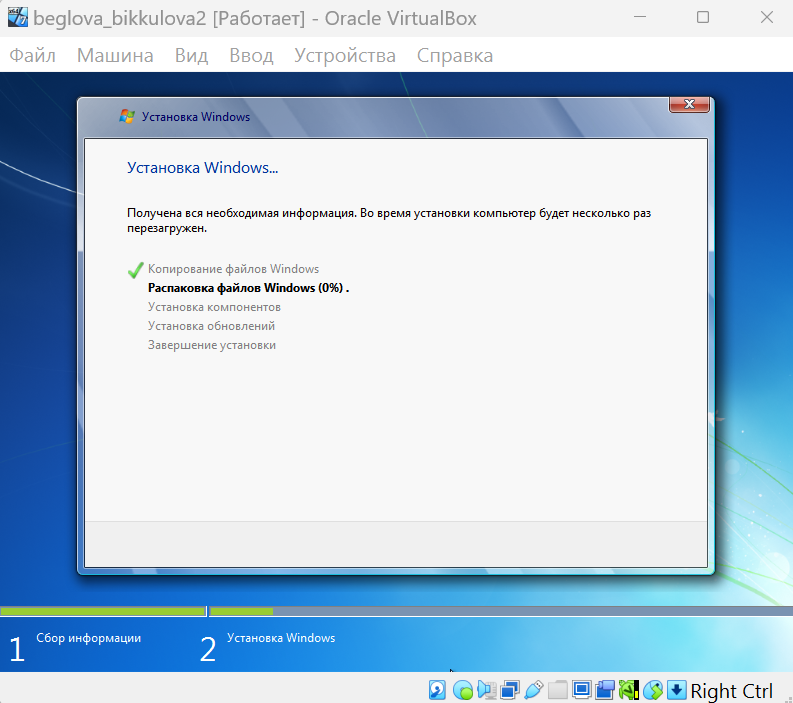


Рисунок 31. Установка ОС на вторую машину

Необходимо убедиться, что машины видят друг друга в локальной сети. Пакеты передаются между машинами без проблем.

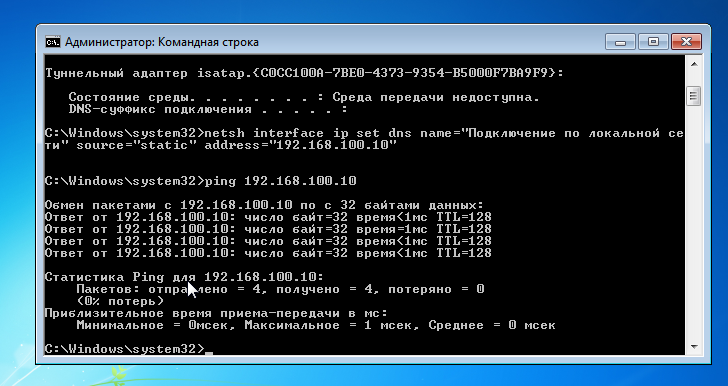


Рисунок 32. Проверка функционирования виртуального сетевого адаптера

Добавим новый компьютер в домен через свойства компьютера в клиентской станции

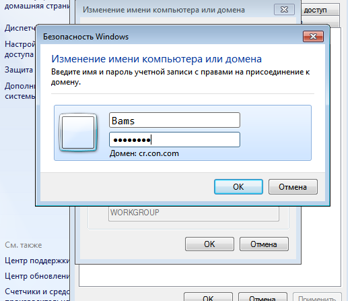


Рисунок 34. Ввод компьютера клиента в домен

После этого необходимо создать нового пользователя на сервере и настроить для него график рабочего времени

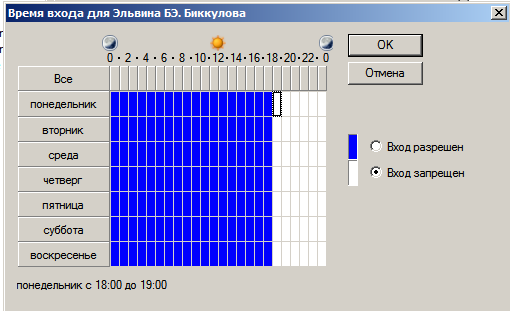


Рисунок 35. Установка расписания входа для пользователя

Добавим к серверу два дополнительных динамических жестких диска и разметим их

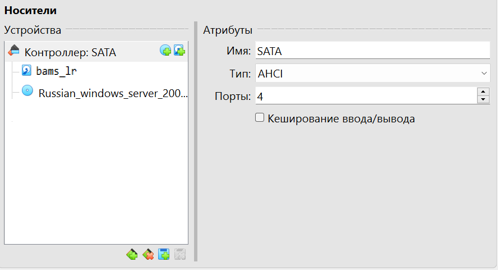
**

Рисунок 36. Добавление жесткого диска

Разметим один диск, создав простой том

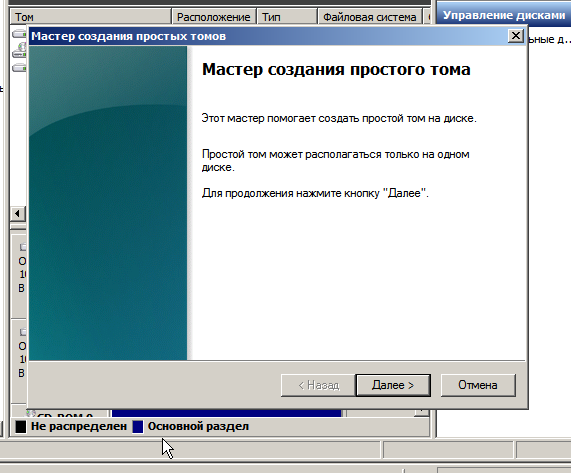


Рисунок 37. Создание простого тона для диска 1

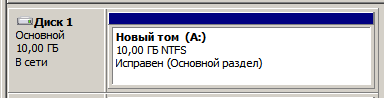


Рисунок 38. Результат разметки диска 1 (А)

Расширим диск 1 за счет диска 2

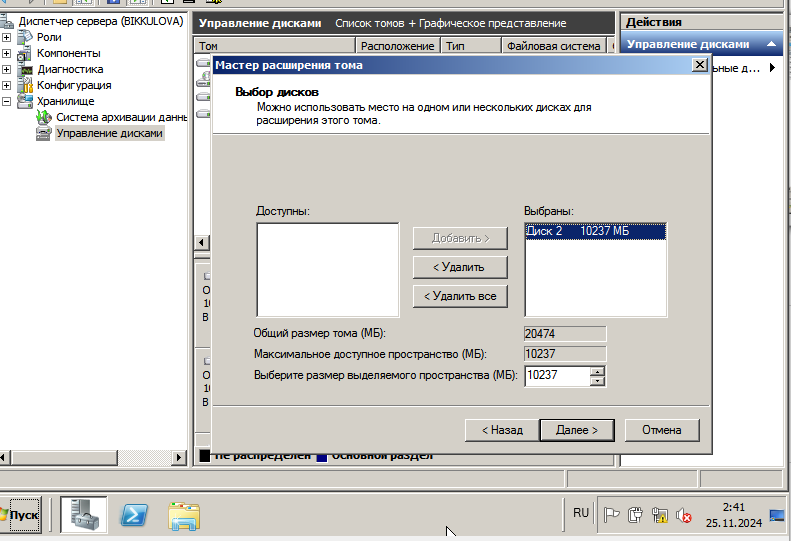


Рисунок 39. Расширение диска 1

В результате получается составной том

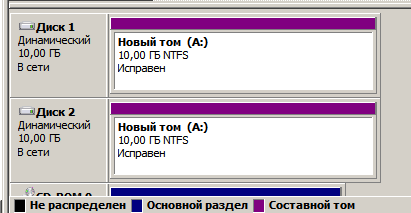


Рисунок 40. Составной том

Далее настроим квоты для пользователей

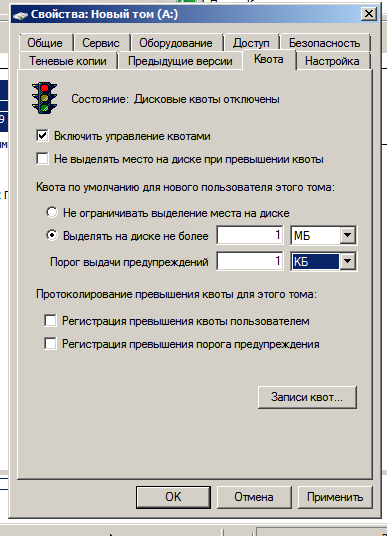


Рисунок 41. Управление дисковой квотой

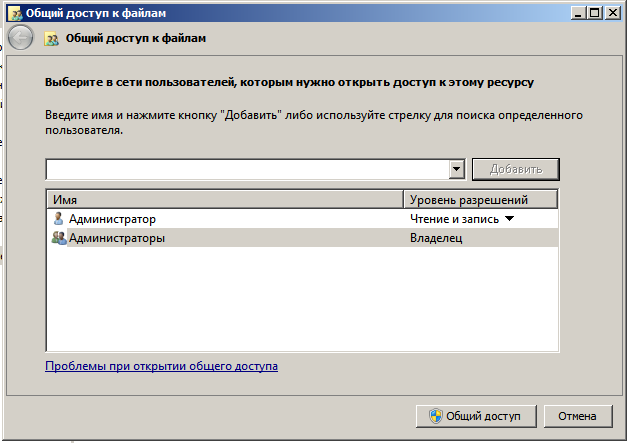


Рисунок 42. Настройка общего доступа к файлам

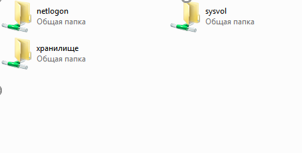


Рисунок 43. Общий доступ сервера для клиента

**Вывод**: Получили практические знания об администрировании файлового сервера в сетях, управляемых ОС Windows, создании доменных учётных записей, работе с ролью файлового сервера, организации квот, управлению общими папками и автономными файлами.

**Лабораторная работа №8. Мониторинг файлового сервера. Автономные файлы. Службы DNS и DHCP.**

**Цель работы**: Получение практических знаний о мониторинге использования дисковой подсистемы сервера, организации и доступности автономных файлов в сети, а также настройках ролей DNS и DHCP в домене.

**Ход работы:**

Установим оснастку «Управление общими ресурсами и хранилищами»

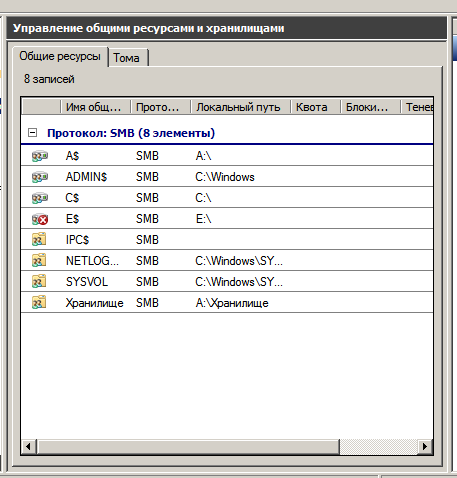


Рисунок 44. Управление общими ресурсами и хранилищами

Необходимо подключить ресурс ADMIN$, опубликованный в общий доступ на клиенте как сетевой диск и добавить службы ролей мониторинга за определенным томом

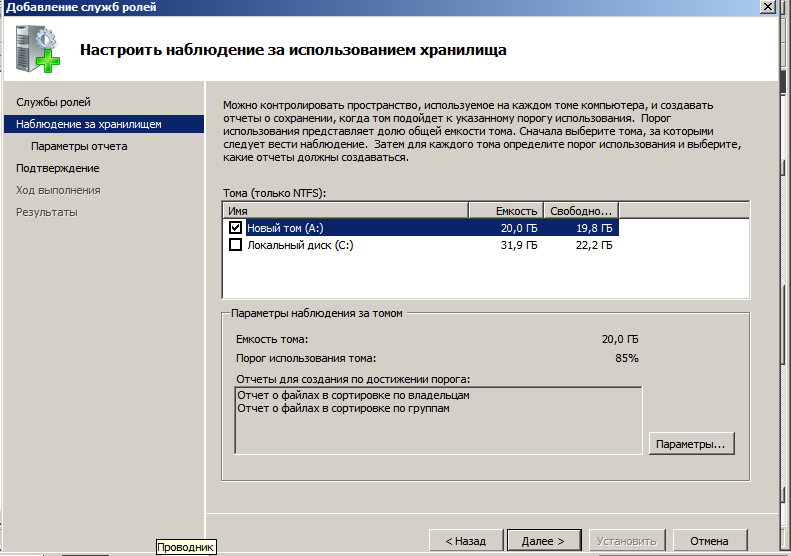


Рисунок 45. Выбор тома А

Создадим квоту для папки «Хранилище» из тома А

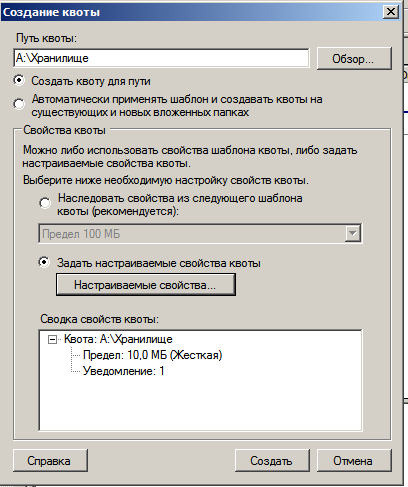


Рисунок 46. Создание квоты

Установим ограничение на типы файлов (изображения)

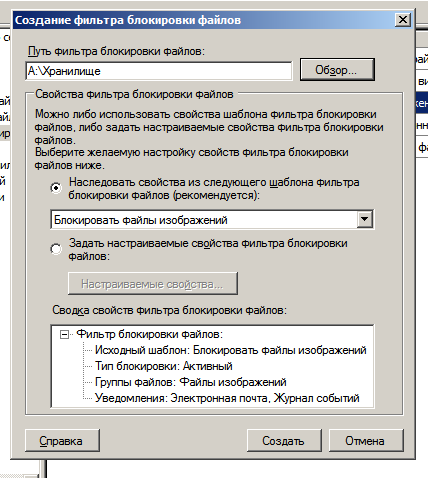


Рисунок 47. Создание фильтра блокировки файлов

Попытаемся добавить в папку изображение

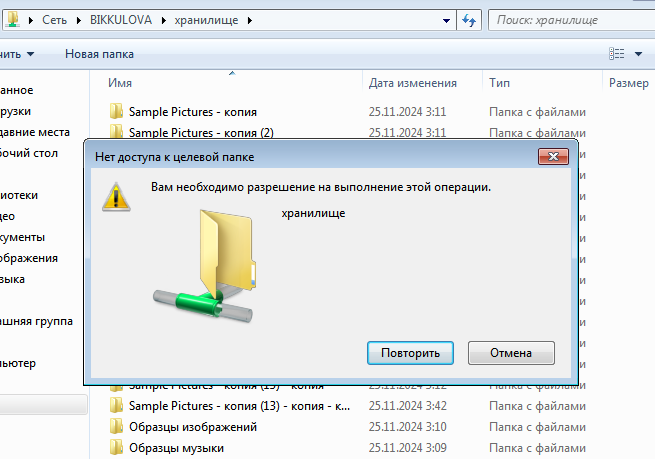


Рисунок 48. Результат работы фильтра

Сгенерируем отчет по использованию пространства на диске

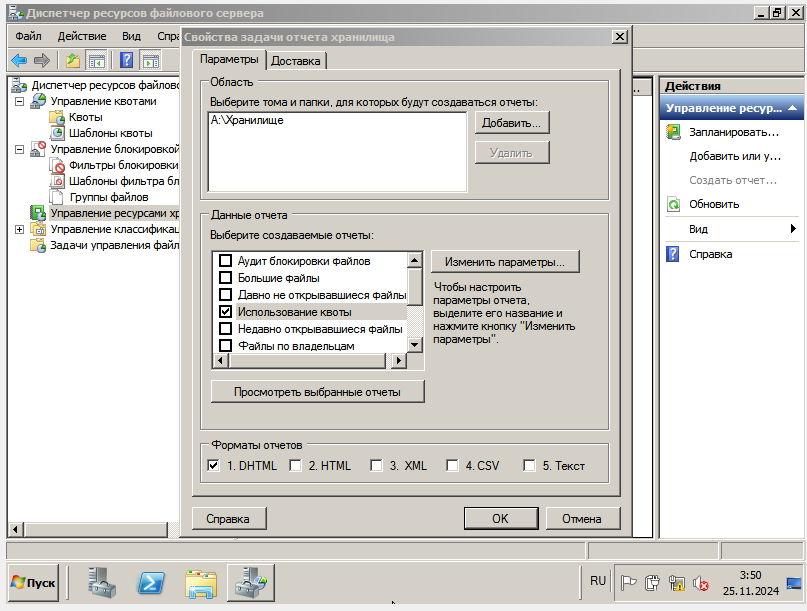


Рисунок 49. Создание отчета

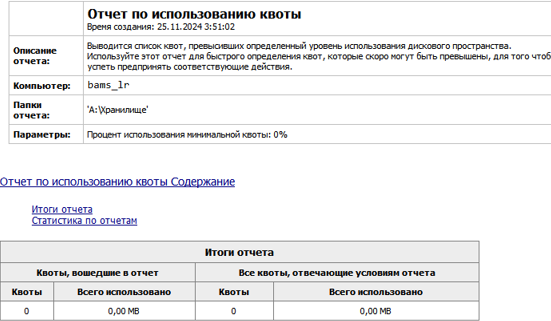


Рисунок 50. Отчет по использованию квоты

Перейдём к понятию и функциональности автономных файлов.

Настроим автономный режим

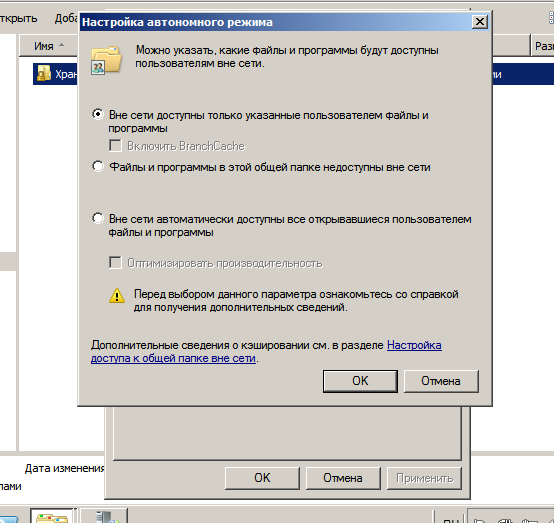


Рисунок 51. Настройка автономного режима

Далее включаем автономность папки Хранилище и проверяем, доступна ли она после отключения адаптера

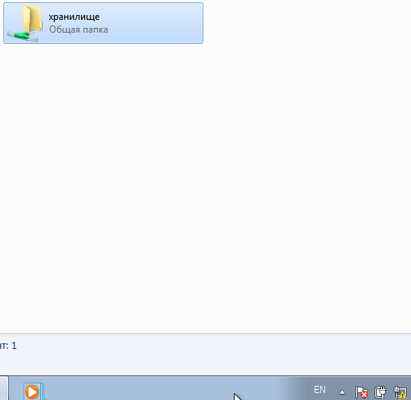


Рисунок 52. Папка после отключения адаптера

Вносим изменения в папку: добавляем еще один текстовый файл, включаем адаптер и проверяем, произошли ли изменения на сервере

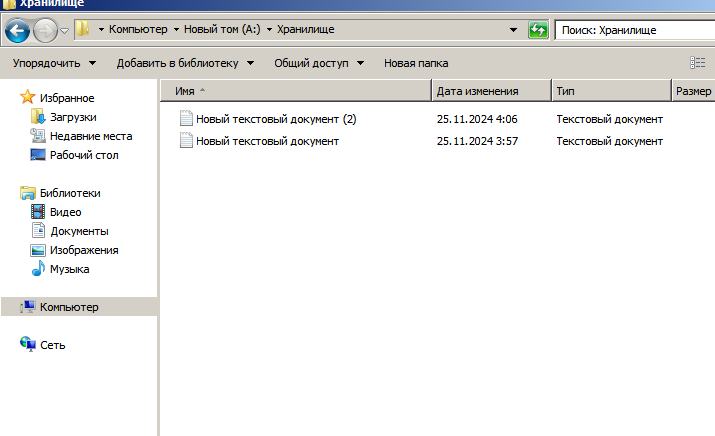


Рисунок 53. Папка на сервере после синхронизации

Далее установим службу DFS

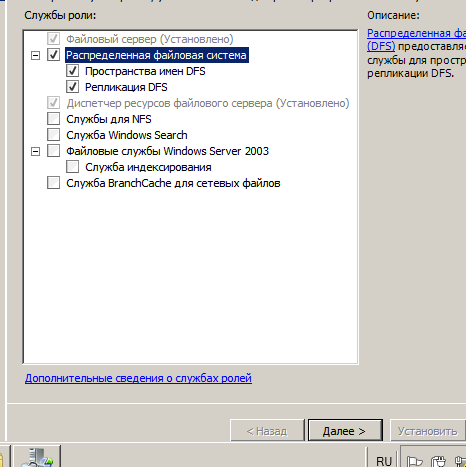


Рисунок 54. Установка DFS

Добавим новый ресурс в пространство имен

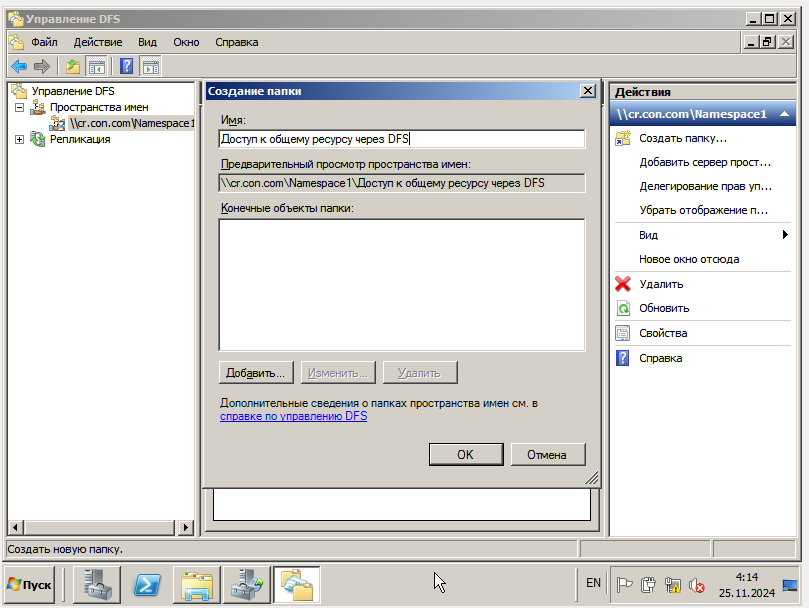


Рисунок 55. Создание папки

Создадим псевдоним домену и проверим это на клиенте

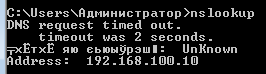


Рисунок 56. Проверка командой nsloopup

Создадим обратное расширение имен

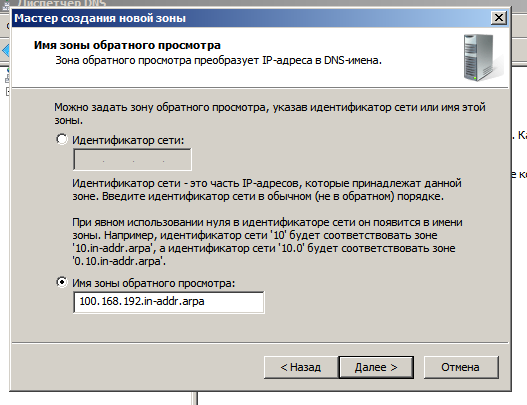


Рисунок 57. Создание новой зоны обратного просмотра

Проверяем на клиенте, все ли верно установлено

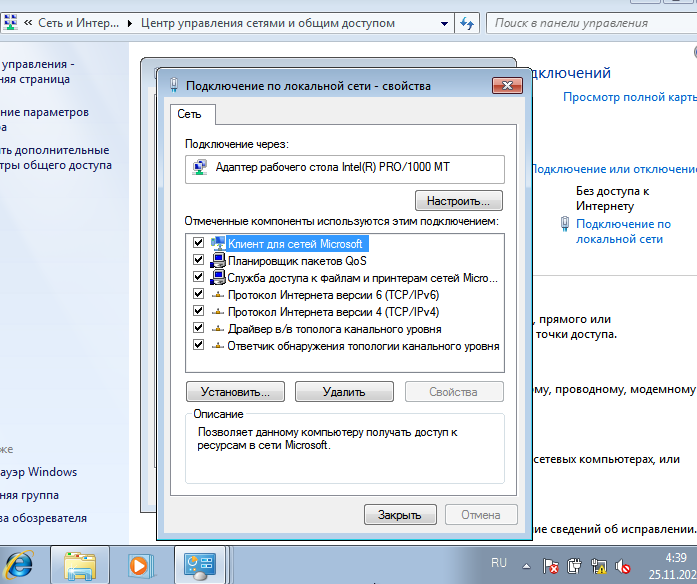


Рисунок 58. Установка получения IP-адреса по DHCP

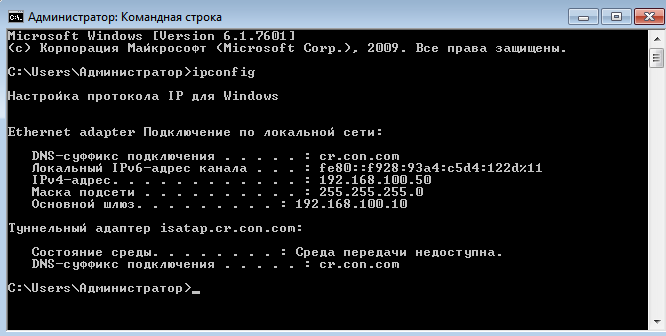


Рисунок 59. Проверка установленного IP-адреса

**Вывод:** Получили практические знания о мониторинге использования дисковой подсистемы сервера, организации и доступности автономных файлов в сети, а также настройках ролей DNS и DHCP в домене.