**Практикум 13**

**Вариант 4**

<https://github.com/VsevolodTilta/Practic>

**Задание 1.**

1. Создать абстрактный класс Trans с методами позволяющим вывести на экран информацию о транспортном средстве, а также определить грузоподъемность транспортного средства.
2. Создать производные классы: Легковая\_машина (марка, номер, скорость, грузоподъемность), Мотоцикл (марка, номер, скорость, грузоподъемность, наличие коляски, при этом если коляска отсутствует, то грузоподъемность равна 0), Грузовик (марка, номер, скорость, грузоподъемность, наличие прицепа, при этом если есть прицеп, то грузоподъемность увеличивается в два раза) со своими методами вывода информации на экран, и определения грузоподъемности.
3. Создать базу (массив) из n машин, вывести полную информацию из базы на экран, а также организовать поиск машин, удовлетворяющих требованиям грузоподъемности.

**Листинг приложения**

abstract class Trans

{

public string model;

public string number;

public double speed;

public double liftPower;

public Trans(string m, string numb, double s, double lif\_pow, bool b = false)

{

model = m;

number = numb;

speed = s;

liftPower = lif\_pow;

}

public abstract void Print();

}

class Moto : Trans

{

public bool kolyaska;

public Moto(string m, double s, string numb, double lif\_pow, bool kol)

: base(m, numb, s, lif\_pow)

{

kolyaska = kol;

if (kol == false)

liftPower = 0;

else

liftPower = lif\_pow;

}

public override void Print()

{

Console.WriteLine("Фирма: {0} \nМаксимальная скорость {1} \nНомер: {2} \nГрузоподьемность: {3} \nНаличие коляски: {4}", model, speed, number, liftPower, kolyaska);

Console.WriteLine();

}

}

class Legko : Trans

{

public Legko(string m, double s, string numb, double lif\_pow)

: base(m, numb, s, lif\_pow) { }

public override void Print()

{

Console.WriteLine("Фирма: {0} \nМаксимальная скорость {1} \nНомер: {2} \nГрузоподьемность: {3} ", model, speed, number, liftPower);

Console.WriteLine();

}

}

class Gruzovie : Trans

{

public bool pricep;

public Gruzovie(string m, double s, string numb, double lif\_pow, bool pric)

: base(m, numb, s, lif\_pow)

{

pricep = pric;

if (pricep == true)

liftPower \*= 2;

}

public override void Print()

{

Console.WriteLine("Фирма: {0} \nМаксимальная скорость {1} \nНомер: {2} \nГрузоподьемность: {3} \nНаличие прицепа: {4}", model, speed, number, liftPower, pricep);

Console.WriteLine();

}

}

class Program

{

static void Main(string[] args)

{

string[] strAss = File.ReadAllLines("Text.txt");

Trans[] mas = new Trans[strAss.Length];

for (int i = 0; i < mas.Length; i++)

{

string[] anal = strAss[i].Split(';');

switch (anal[0])

{

case "Moto":

{

mas[i] = new Moto(anal[1], double.Parse(anal[2]), anal[3], double.Parse(anal[4]), bool.Parse(anal[5]));

break;

}

case "Legko":

{

mas[i] = new Legko(anal[1], double.Parse(anal[2]), anal[3], double.Parse(anal[4]));

break;

}

case "Gruzovie":

{

mas[i] = new Gruzovie(anal[1], double.Parse(anal[2]), anal[3], double.Parse(anal[4]), bool.Parse(anal[5]));

break;

}

}

}

for (int i = 0; i < mas.Length; i++)

mas[i].Print();

Console.Write("Введите грузоподъемность: ");

int gruz = int.Parse(Console.ReadLine());

for (int i = 0; i < mas.Length; i++)

{

if (gruz <= mas[i].liftPower)

mas[i].Print();

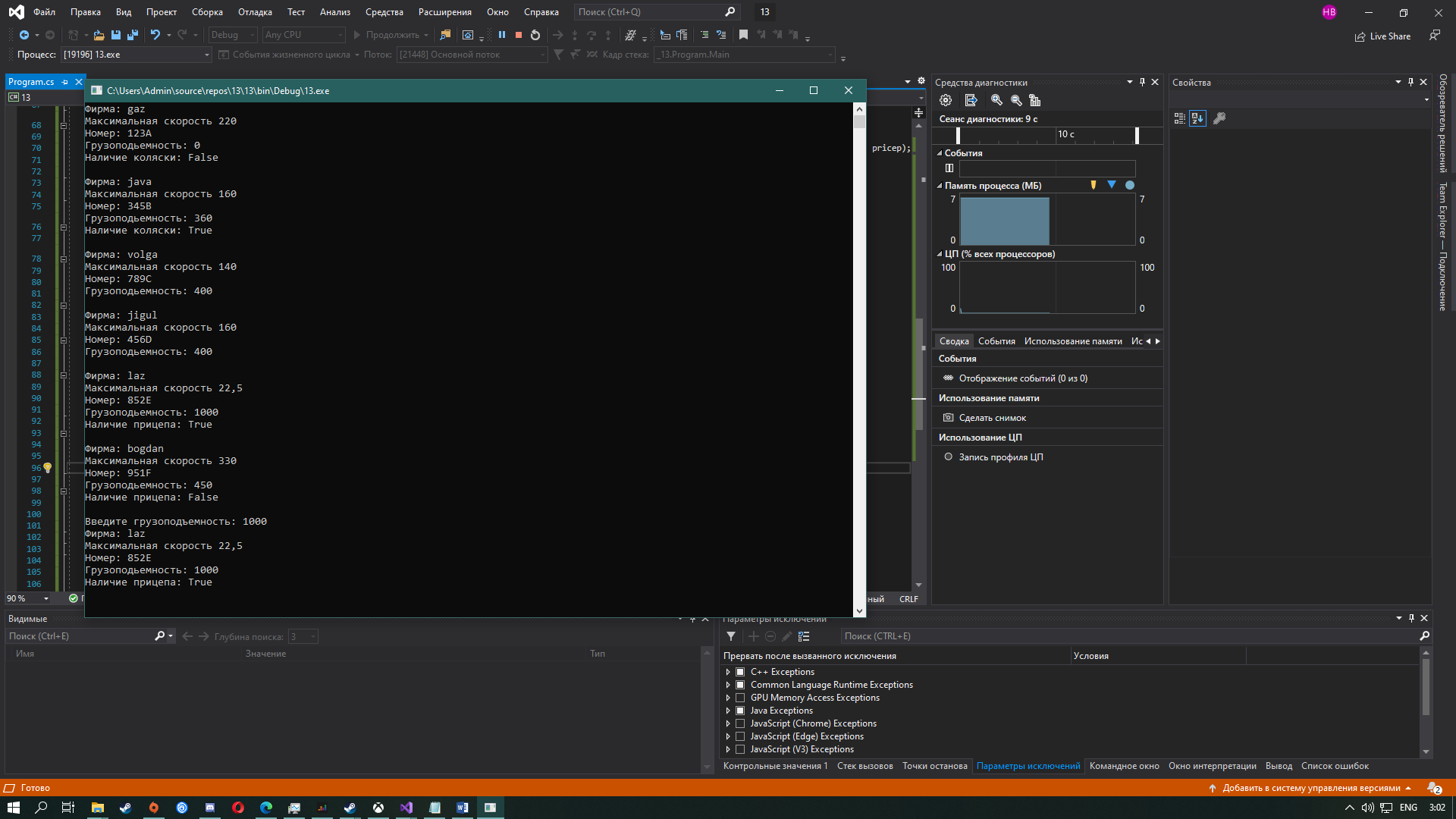
}

Console.ReadKey();

}

}

**Консольное приложение**



**Листинг приложения windows forms**

public partial class Form1 : Form

{

Trans[] mas;

public Form1()

{

InitializeComponent();

}

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

{

try

{

int gruz = int.Parse(textBox1.Text);

for (int i = 0; i < mas.Length; i++)

{

if (gruz <= mas[i].liftPower)

mas[i].Print(richTextBox2);

}

}

catch

{

MessageBox.Show("Неверные данные");

}

}

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

{

string[] strAss = File.ReadAllLines("Text.txt");

mas = new Trans[strAss.Length];

for (int i = 0; i < mas.Length; i++)

{

string[] anal = strAss[i].Split(';');

switch (anal[0])

{

case "Moto":

{

mas[i] = new Moto(anal[1], double.Parse(anal[2]), anal[3], double.Parse(anal[4]), bool.Parse(anal[5]));

break;

}

case "Legko":

{

mas[i] = new Legko(anal[1], double.Parse(anal[2]), anal[3], double.Parse(anal[4]));

break;

}

case "Gruzovie":

{

mas[i] = new Gruzovie(anal[1], double.Parse(anal[2]), anal[3], double.Parse(anal[4]), bool.Parse(anal[5]));

break;

}

}

}

for (int i = 0; i < mas.Length; i++)

mas[i].Print(richTextBox1);

}

abstract class Trans

{

public string model;

public string number;

public double speed;

public double liftPower;

public Trans(string m, string numb, double s, double lif\_pow, bool b = false)

{

model = m;

number = numb;

speed = s;

liftPower = lif\_pow;

}

public abstract void Print(RichTextBox ok);

}

class Moto : Trans

{

public bool kolyaska;

public Moto(string m, double s, string numb, double lif\_pow, bool kol)

: base(m, numb, s, lif\_pow)

{

kolyaska = kol;

if (kol == false)

liftPower = 0;

else

liftPower = lif\_pow;

}

public override void Print(RichTextBox ok)

{

ok.Text += $"Фирма: {model} \nМаксимальная скорость {speed} \nНомер: {number} \nГрузоподьемность: {liftPower} \nНаличие коляски: {kolyaska} \n";

}

}

class Legko : Trans

{

public Legko(string m, double s, string numb, double lif\_pow)

: base(m, numb, s, lif\_pow) { }

public override void Print(RichTextBox ok)

{

ok.Text += $"Фирма: {model} \nМаксимальная скорость {speed} \nНомер: {number} \nГрузоподьемность: {liftPower} \n";

}

}

class Gruzovie : Trans

{

public bool pricep;

public Gruzovie(string m, double s, string numb, double lif\_pow, bool pric)

: base(m, numb, s, lif\_pow)

{

pricep = pric;

if (pricep == true)

liftPower \*= 2;

}

public override void Print(RichTextBox ok)

{

ok.Text += $"Фирма: {model} \nМаксимальная скорость {speed} \nНомер: {number} \nГрузоподьемность: {liftPower} \nНаличие прицепа: {pricep} \n";

}

**Приложение windows forms**

