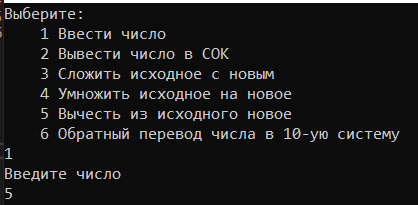
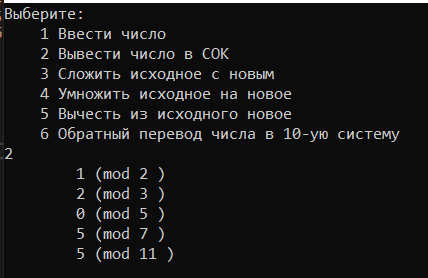
Скобелкин Г.К., ФИб-3301.

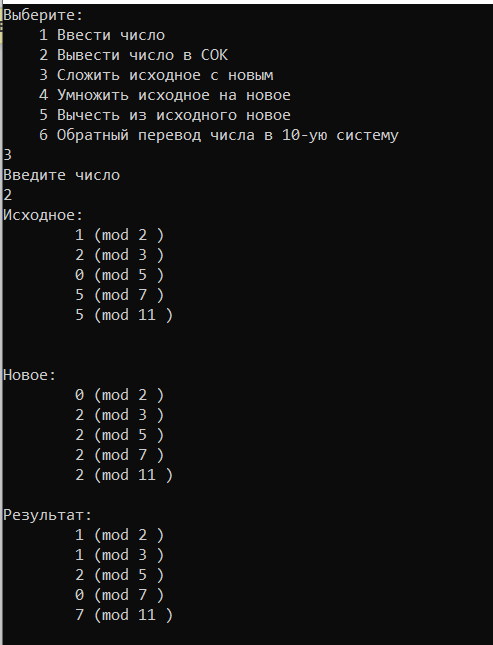
Лабораторная работа по теме «Класс вычетов».

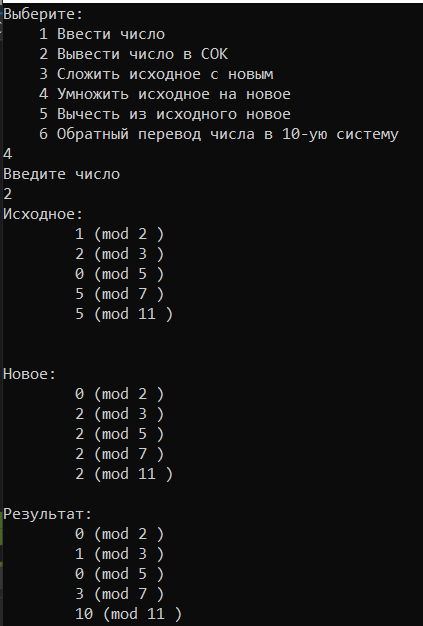
Задание: реализовать класс Система классов, который умеет: переводить число в СОК, складывать, умножать, вычитать, делать обратный перевод.

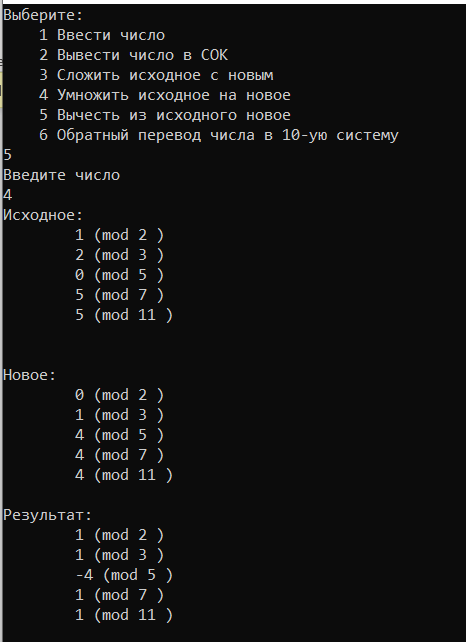
Результат работы программы:

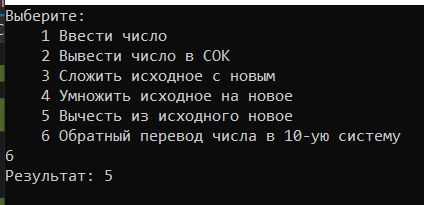
Ввёл число:  


Вывел число:  


Сложение:  


Умножение:  


Вычитание:  


Перевод числа обратно в «нормальный» вид:  


Код программы:

Основной класс:

using System;

namespace LAb6\_SysClas\_

{

class Program

{

public static string menu =

($@"Выберите:

1 Ввести число

2 Вывести число в СОК

3 Сложить исходное с новым

4 Умножить исходное на новое

5 Вычесть из исходного новое

6 Обратный перевод числа в 10-ую систему");

static void Main(string[] args)

{

SystemModulClass num = new SystemModulClass();

int t = 1;

while(t>0)

{

Console.Clear();

Console.WriteLine( menu );

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

switch (t)

{

case 1:

Console.WriteLine("Введите число");

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

num = new SystemModulClass(a);

break;

case 2:

Console.WriteLine(num);

break;

case 3:

Console.WriteLine("Введите число");

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

Console.WriteLine("Исходное:\n" + num + "\n");

SystemModulClass bnum = new SystemModulClass(b);

Console.WriteLine("Новое:\n" + bnum);

Console.WriteLine("Результат:\n" + (num+bnum));

break;

case 4:

Console.WriteLine("Введите число");

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

Console.WriteLine("Исходное:\n" + num + "\n");

bnum = new SystemModulClass(b);

Console.WriteLine("Новое:\n" + bnum);

Console.WriteLine("Результат:\n" + (num \* bnum));

break;

case 5:

Console.WriteLine("Введите число");

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

Console.WriteLine("Исходное:\n" + num + "\n");

bnum = new SystemModulClass(b);

Console.WriteLine("Новое:\n" + bnum);

Console.WriteLine("Результат:\n" + (num - bnum));

break;

case 6:

Console.WriteLine("Результат: " + num.ReTransletion());

break;

default:

break;

}

Console.ReadKey();

}

}

}

}

Система классов:  
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.IO;

namespace LAb6\_SysClas\_

{

public class SystemModulClass

{

// массив простых чисел

public List<int> simple = new List<int>();

public List<Pair> sv { get; set; }

// конструкторы

public SystemModulClass()

{

sv = new List<Pair>();

}

public SystemModulClass(int number)

{

int i = 0;

simple = Create();

sv = new List<Pair>();

while (i < 5)

{

sv.Add(new Pair(number % simple[i], simple[i]));

i++;

}

}

// сложение

static public SystemModulClass operator +(SystemModulClass a, SystemModulClass b)

{

SystemModulClass res = new SystemModulClass();

for (int i = 0; i < a.sv.Count; i++)

res.sv.Add(a.sv[i] + b.sv[i]);

return res;

}

// вычитание

static public SystemModulClass operator -(SystemModulClass a, SystemModulClass b)

{

SystemModulClass res = new SystemModulClass();

for (int i = 0; i < a.sv.Count; i++)

res.sv.Add(a.sv[i] - b.sv[i]);

return res;

}

// умножение

static public SystemModulClass operator \*(SystemModulClass a, SystemModulClass b)

{

SystemModulClass res = new SystemModulClass();

for (int i = 0; i < a.sv.Count; i++)

res.sv.Add(a.sv[i] \* b.sv[i]);

return res;

}

// перевод числа в обычный вид

public int ReTransletion()

{

List<int> mi = new List<int>();

List<int> mi\_obr = new List<int>();

int CalcM()

{

int r = 1;

foreach (var item in sv)

r \*= item.m;

return r;

}

int M = CalcM();

int Inverse(int a, int n)

{

int d = 0, x = 1, y = 0;

NOD(a, n, ref x, ref y, ref d);

if (d == 1)

return x;

return 0;

}

foreach (var item in sv) //находим Mi и обратные

{

mi.Add(M / item.m);

mi\_obr.Add(Inverse(mi.Last(), item.m));

}

int res = 0;

for (int i = 0; i < mi.Count; i++)

res += mi[i] \* mi\_obr[i] \* sv[i].a;

return res % M;

}

// нод

public static void NOD(int a, int b, ref int x, ref int y, ref int d)

{

int q, r, x1, x2, y1, y2;

if (b == 0)

{

d = a;

x = 1;

y = 0;

return;

}

x1 = 0;

x2 = 1;

y1 = 1;

y2 = 0;

while (b > 0)

{

q = a / b;

r = a - q \* b;

x = x2 - q \* x1;

y = y2 - q \* y1;

a = b;

b = r;

x2 = x1;

x1 = x;

y2 = y1;

y1 = y;

}

d = a;

x = x2;

y = y2;

}

// заполняет массив простых чисел

public static List<int> Create()

{

string s;

using (StreamReader str = new StreamReader("init.txt", Encoding.Default))

{

s = str.ReadToEnd();

}

string[] f = s.Split();

List<int> res = new List<int>();

foreach (var item in f)

{

res.Add(int.Parse(item));

}

return res;

}

// перевод в string

public override string ToString()

{

string s = "";

foreach (var item in sv)

{

s += $"\t{item.a} (mod {item.m} )\n";

}

return s;

}

}

}

namespace LAb6\_SysClas\_

{

public class Pair

{

public int a { get; set; }

public int m { get; set; }

public Pair(int \_a, int \_b)

{

a = \_a;

m = \_b;

}

public Pair()

{

a = 0;

m = 0;

}

static public Pair operator +(Pair a,Pair b) => new Pair((a.a + b.a) % b.m, b.m);

static public Pair operator \*(Pair a, Pair b) => new Pair((a.a \* b.a) % b.m, b.m);

static public Pair operator -(Pair a, Pair b) => new Pair((a.a - b.a) % b.m, b.m); // a больше

}

}