Assignment 4

//NumTemplate.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Assignment4

{

abstract class NumTemplate

{

public int no = 0;

public NumTemplate()

{

this.no = 2;

}

public NumTemplate(int value)

{

this.no = value;

}

public abstract void Accept();

public abstract void Display();

public abstract bool ChkEven();

public abstract void DisplayFactors();

public abstract void SumFactors();

}

}

//NumOperation.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Assignment4

{

class NumOperation : NumTemplate

{

public int iNo, iSum;

public override void Accept()

{

Console.Write("Enter the Number : ");

iNo = Convert.ToInt32(Console.ReadLine());

if (iNo < 0)

{

iNo = -iNo;

}

}

public override bool ChkEven()

{

if ((iNo % 2) == 0)

{

return true;

}

else

{

return false;

}

}

public override void Display()

{

bool bRet = true;

bRet = ChkEven();

Console.Write("Check Even Or NOT : \n");

if (bRet == true)

{

Console.WriteLine(" Number is Even Number");

}

else

{

Console.WriteLine(" Number is Not Even Number");

}

Console.Write("Addition is{0}", iSum);

}

public override void DisplayFactors()

{

int i = 0;

Console.Write("Factor is ", iNo);

for (i = 1; i <= (iNo / 2); i++)

{

if ((iNo % i) == 0)

{

Console.Write("{0}, ", i);

}

}

Console.Write("{0}", iNo);

}

public override void SumFactors()

{

int i = 0;

for (i = 1; i <= (iNo / 2); i++)

{

if ((iNo % i) == 0)

{

iSum = iSum + i;

}

}

iSum = iSum + iNo;

}

}

}

//NumberActivity.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Assignment4

{

class NumberActivity : NumOperation

{

public int iFact;

public NumOperation Num;

public NumberActivity()

{

Num = new NumOperation();

Num.Accept();

iNo = Num.iNo;

}

public NumberActivity(NumOperation obj)

{

iNo = obj.iNo;

iFact = 1;

}

public void Factorial()

{

int i = 0;

for (i = 1; i <= iNo; i++)

{

iFact = iFact \* i;

}

}

public bool ChkPrime()

{

int i = 0, chk = 0;

for (i = 2; i <= iNo / 2; i++)

{

if ((iNo % i) == 0)

{

chk = 1;

break;

}

}

if (chk == 0)

{

return true;

}

else

{

return false;

}

}

public bool ChkPerfect()

{

int sum = 0, i = 0;

for (i = 1; i <= iNo / 2; i++)

{

if ((iNo % i) == 0)

{

sum = sum + i;

}

}

if (sum == iNo)

{

return true;

}

else

{

return false;

}

}

public void Display()

{

Console.WriteLine("Factorial of Number{0}", iFact);

}

}

}

//Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Assignment4

{

class Program

{

static void Main(string[] args)

{

bool prime, perfect;

NumOperation ob = new NumOperation();

ob.Accept();

ob.ChkEven();

ob.SumFactors();

ob.Display();

ob.DisplayFactors();

NumberActivity obj = new NumberActivity(ob);

obj.Factorial();

obj.Display();

prime = obj.ChkPrime();

if (prime == true)

{

Console.WriteLine(" Number is Prime Number");

}

else

{

Console.WriteLine(" Number is Not a Prime Number");

}

perfect = obj.ChkPerfect();

if (perfect == true)

{

Console.WriteLine(" Number is Perfect Number");

}

else

{

Console.WriteLine(" Number is Not a Perfect Number");

}

}

}

}