Решение задач

1. В бинарном файле записаны целые числа. Найти количество простых чисел в файле. Найти самое большое простое число и записать его в конец файла.

namespace SimpleNumbers

{

class Program

{

static int ReadNumber(string message, int left, int right)

{

Console.WriteLine(message);

int number;

bool ok;

do

{

ok = Int32.TryParse(Console.ReadLine(), out number);

ok = ok && number >= left && number < right;

} while (!ok);

return number;

}

static bool IsSimple(int number)

{

if (number <2) return false;

int x = 2;

for (; x < Math.Sqrt(number); x++)

if (number % x == 0) return false;

return true;

}

static int CountSimple(string filename)

{

int count = 0;

int number;

try

{

using (FileStream f = new FileStream(filename, FileMode.Open))

{

using (BinaryReader bf = new BinaryReader(f))

try

{

while (true)

{

number = bf.ReadInt32();

Console.Write(number + "\t");

if (IsSimple(number)) count++;

}

}

catch (EndOfStreamException)

{

Console.WriteLine();

}

}

}

catch (Exception e)

{

Console.WriteLine("Error: " + e.Message);

}

return count;

}

static int MaxSimple(string filename)

{

int number;

int max = Int32.MinValue;

try

{

FileStream f = new FileStream(filename, FileMode.Open);

BinaryReader bf = new BinaryReader(f);

try

{

while (true)

{

number = bf.ReadInt32();

if (IsSimple(number))

if (max < number) max=number;

}

}

catch (EndOfStreamException)

{

Console.WriteLine();

bf.Close();

f.Close();

}

}

catch (Exception e)

{

Console.WriteLine("Error: " + e.Message);

}

return max;

}

static void WriteMaxSimple(string filename)

{

int max = MaxSimple(filename);

if(max==Int32.MaxValue)

{

Console.WriteLine("Нет простых чисел");

return;

}

try

{

FileStream f = new FileStream(filename, FileMode.Append);//в конец

BinaryWriter bf = new BinaryWriter(f);

try

{

bf.Write(max);

}

catch (Exception e)

{

Console.WriteLine("Error: " + e.Message);

}

finally

{

bf.Close();

f.Close();

}

}

catch(FileNotFoundException)

{

Console.WriteLine("Файл с таким именем не найден");

}

}

static void ReadFile(string filename)

{

int number;

try

{

FileStream f = new FileStream(filename, FileMode.Open);

BinaryReader bf = new BinaryReader(f);

try

{

while (true)

{

number = bf.ReadInt32();

Console.Write(number + "\t");

}

}

catch (Exception) //ошибка при чтении

{

Console.WriteLine();

}

finally

{

bf.Close();

f.Close();

}

}

catch (EndOfStreamException) //такого файла нет

{

Console.WriteLine();

}

}

static void WriteFile(string filename, int size)

{

FileStream f = new FileStream(filename, FileMode.Create);

BinaryWriter bf = new BinaryWriter(f);

Random rnd = new Random();

int number;

try

{

for (int i = 0; i < size; i++)

{

number = rnd.Next(100);

bf.Write(number);

}

}

catch (Exception e)

{

Console.WriteLine("Error: " + e.Message);

}

finally

{

bf.Close();

f.Close();

}

}

static void Main(string[] args)

{

int number = ReadNumber("Сколько чисел записать в файл?", 0, 100);

Console.WriteLine("Введите имя файла");

string filename=Console.ReadLine();

WriteFile(filename,number);

Console.WriteLine(CountSimple(filename));

WriteMaxSimple(filename);

ReadFile(filename);

}

}

}

1. В текстовом файле записаны целые числа. Отсортировать их по возрастанию и записать другой тестовый файл.

class Program

{

static int[] MakeRndMas(int size)

{

Random rnd = new Random();

int[] mas = new int[size];

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

mas[i] = rnd.Next(-100, 100);

return mas;

}

static void MakeTextFile(string filename, int[]mas)

{

using (StreamWriter f = new StreamWriter(filename))

{

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

{

f.Write(mas[i] + " ");

}

}

}

static string ReadTextFile(string filename)

{

string s;

using (StreamReader f = new StreamReader(filename))

s = f.ReadToEnd();

return s;

}

static int[] MakeArray(string s)

{

string[] strArr = s.Split();

int[] intArr = new int[strArr.Length-1];

for(int i=0;i<strArr.Length-1;i++)

{

intArr[i] = Int32.Parse(strArr[i]);

}

return intArr;

}

static void PrintFile(string filename)

{

string s;

using (StreamReader f = new StreamReader(filename))

s = f.ReadToEnd();

Console.WriteLine(s);

}

static void AddToEnd(string filename, int[]mas)

{

FileStream f = new FileStream(filename, FileMode.Append);

StreamWriter textFile = new StreamWriter(f);

textFile.WriteLine("\nОтсортированный массив:");

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

{

textFile.Write(mas[i] + " ");

}

textFile.Close();

f.Close();

}

static void Main(string[] args)

{

string filename1 = "../text1.txt", filename2 = "../text2.txt";

Console.WriteLine("Сколько элементов будет в массиве?");

int size = Int32.Parse(Console.ReadLine());

int[] mas1 = MakeRndMas(size);

MakeTextFile(filename1, mas1);

PrintFile(filename1);

string s = ReadTextFile(filename1);

int[] mas2 = MakeArray(s);

Array.Sort(mas2);

MakeTextFile(filename2, mas2);

PrintFile(filename2);

Console.WriteLine("Запись в конец файла");

AddToEnd(filename1, mas2);

PrintFile(filename1);

}

}

1. Удалить из бинарного файла, в который записаны данные типа Person запись с номером k.

[Serializable]

struct Person

{

public string name;

public int age;

public Person(string N="", int A=0)

{

name = N; age = A;

}

public void Show()

{

Console.WriteLine(name + ", " + age );

}

public void Input()

{

Console.Write("Имя:"); name = Console.ReadLine();

Console.Write("Возраст"); age = Convert.ToInt32(Console.ReadLine());

}

}

class Program

{

static string filename = @"../bd.bin";

static Person[] MakePersonArray(int size)

{

List<Person> list = new List<Person>();

for (int i = 0; i < size; i++)

{

Person p = new Person();

p.Input();

list.Add(p);

}

return list.ToArray<Person>();

}

static void CreateBD(Person[] arr)

{

BinaryFormatter bf = new BinaryFormatter();

using (FileStream fs = new FileStream(filename, FileMode.Create))

foreach (Person p in arr)

bf.Serialize(fs, p);

Console.WriteLine("Объект сериализован");

}

static Person[] ReadBD()

{

Person[] arr;

List<Person> list = new List<Person>();

BinaryFormatter bf = new BinaryFormatter();

using (FileStream fs = new FileStream(filename, FileMode.Open))

{

// arr=(Person[])bf.Deserialize(fs);

try

{

while (true)

{

Person p = (Person)bf.Deserialize(fs);

list.Add(p);

}

}

catch (Exception )

{

}

}

return list.ToArray();

}

static void PrintArray(Person[]arr)

{

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

arr[i].Show();

}

static void DeleteRecord(int number)

{

Person[] arr;

BinaryFormatter bf = new BinaryFormatter();

List<Person> list = new List<Person>();

using (FileStream fs = new FileStream(filename, FileMode.Open))

{

using (FileStream temp = new FileStream("temp", FileMode.Create))

{

// arr = (Person[])bf.Deserialize(fs);

try

{

while (true)

{

Person p = (Person)bf.Deserialize(fs);

list.Add(p);

}

}

catch (Exception)

{

}

arr = list.ToArray();

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

if (i != number - 1)

bf.Serialize(temp, arr[i]);

}

}

FileInfo file1 = new FileInfo(filename);

file1.Delete();

FileInfo file2 = new FileInfo("temp");

file2.MoveTo(filename);

}

static void Main(string[] args)

{

Person[] arr = MakePersonArray(3);

CreateBD(arr);

arr = ReadBD();

PrintArray(arr);

Console.WriteLine("Удаление записи");

DeleteRecord(2);

arr = ReadBD();

PrintArray(arr);

}

}