

Инструкция по работе с тестирующей программой (TS.exe)

Общие положения

Задания на олимпиадах по программированию предполагают решение в виде написания программы на каком-либо языке программирования. Разработанные программы должны выполнять обработку каких-либо входных данных согласно условию задачи. Так как в общем случае проверить корректность алгоритма, предложенного участниками олимпиады в своих решениях невозможно, а тем более невозможно сравнить, какое из решений «более правильно», то для оценки решения общепринято использовать тесты.

Тесты для задачи представляет собой наборы входных данных согласно условию задачи, а также тестирующий модуль, который для конкретного набора входных данных (теста) может проверить правильность набора выходных данных, полученных с помощью программы-решения участника олимпиады. Для каждого задания на олимпиаде разрабатывается свой набор тестов.

Оценка решения участника производится в зависимости от количества пройденных решением тестов. Тест считается пройденным, если для данного теста с помощью решения участника получены выходные данные, правильные согласно условию задачи. Каждому тесту сопоставляется какое-то количество баллов за прохождение теста в зависимости от сложности данного теста. Таким образом, каждое решение оценивается в сумму баллов за пройденные данным решением тесты.

Тестирующая программа (TS.exe) предназначена для тестирования решений участников олимпиад по программированию с помощью тестов.

Правила оформления решений

Учитывая, что решения участников олимпиады оцениваются с помощью тестов, при разработке решений следует строго соблюдать форматы входных и выходных данных, указанные в условии задачи.

В общем случае программа-решение состоит из трех частей:

- чтение входных данных из входного файла;
- обработка данных для поиска правильного ответа;
- запись полученного ответа в выходной файл.

Ниже приводятся примеры оформления решений задачи «Task 0. Сумма чисел» (см. условия задач) на различных языках программирования.

Pascal (Turbo Pascal, Delphi)

```
program Task0;

const
  InputFileName = 'input.txt';
  OutputFileName = 'output.txt';

var
  A, B, C: Integer;
  Input, Output: Text;

begin
  Assign(Input, InputFileName);
  Reset(Input);
  Assign(Output, OutputFileName);
  Rewrite(Output);
```

```
ReadLn(Input, A, B);

C := A + B;

WriteLn(Output, C);

Close(Input);
Close(Output);
end.
```

Для языка Delphi желательно также добавить директиву компилятора {\$APPTYPE CONSOLE} в начале текста программы.

C (Turbo C, Visual C)

```
#include <stdio.h>

const char
    *inputFileName = "input.txt",
    *outputFileName = "output.txt";

void main() {
    int a, b, c;

    freopen(inputFileName, "r", stdin);
    freopen(outputFileName, "w", stdout);

    scanf("%d %d", &a, &b);

    c = a + b;

    printf("%d\n", c);

    fclose(stdin);
    fclose(stdout);
}
```

C++ (Visual C++)

```
#include <fstream>

using namespace std;

const string
    inputFileName = "input.txt",
    outputFileName = "output.txt";

void main() {
    ifstream input(inputFileName.c_str());
    ofstream output(outputFileName.c_str());

    int a, b;
    input >> a >> b;

    int c = a + b;

    output << c << endl;

    input.close();
}
```

```
    output.close();  
}
```

C#

```
using System;  
using System.IO;  
  
class Task0  
{  
    public const string  
        InputFileName = "input.txt",  
        OutputFileName = "output.txt";  
  
    static void Main(string[] args)  
    {  
        StreamReader input = new StreamReader(InputFileName);  
        StreamWriter output = new StreamWriter(OutputFileName);  
  
        string[] parts = input.ReadLine().Split();  
        int a = int.Parse(parts[0]),  
            b = int.Parse(parts[1]);  
  
        int c = a + b;  
  
        output.WriteLine(c);  
  
        input.Close();  
        output.Close();  
    }  
}
```

Java

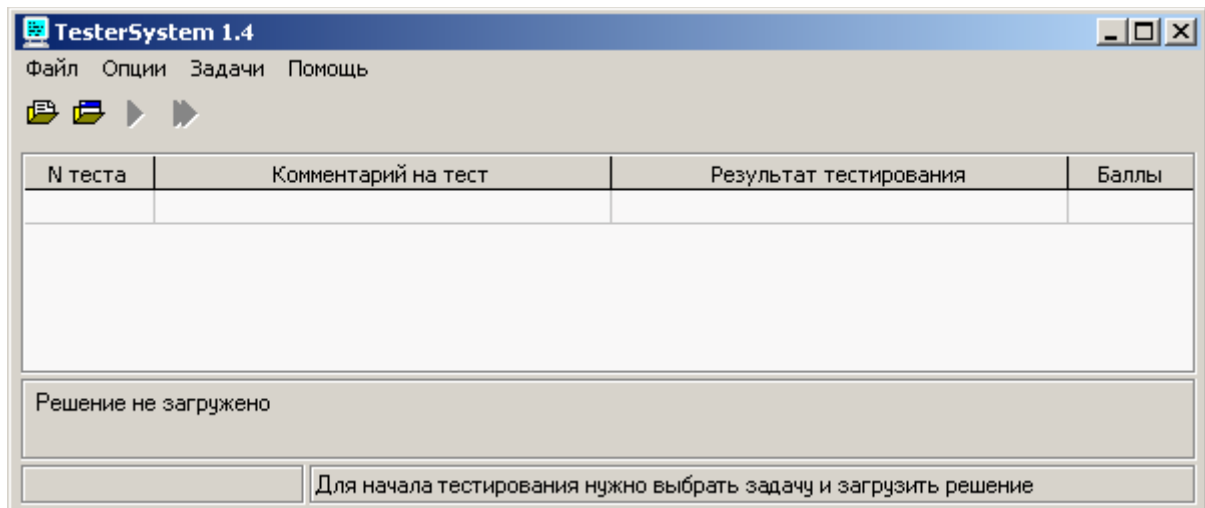
```
import java.util.*;  
import java.io.*;  
  
public class Task0  
{  
    public static final String  
        InputFileName = "input.txt",  
        OutputFileName = "output.txt";  
  
    public static void main(String[] args) throws Exception  
    {  
        Scanner input = new Scanner(new File(InputFileName));  
        PrintWriter output = new PrintWriter(new File(OutputFileName));  
  
        int a = input.nextInt();  
        int b = input.nextInt();  
  
        int c = a + b;  
  
        output.println(c);  
  
        input.close();  
        output.close();  
    }  
}
```


Внимание!

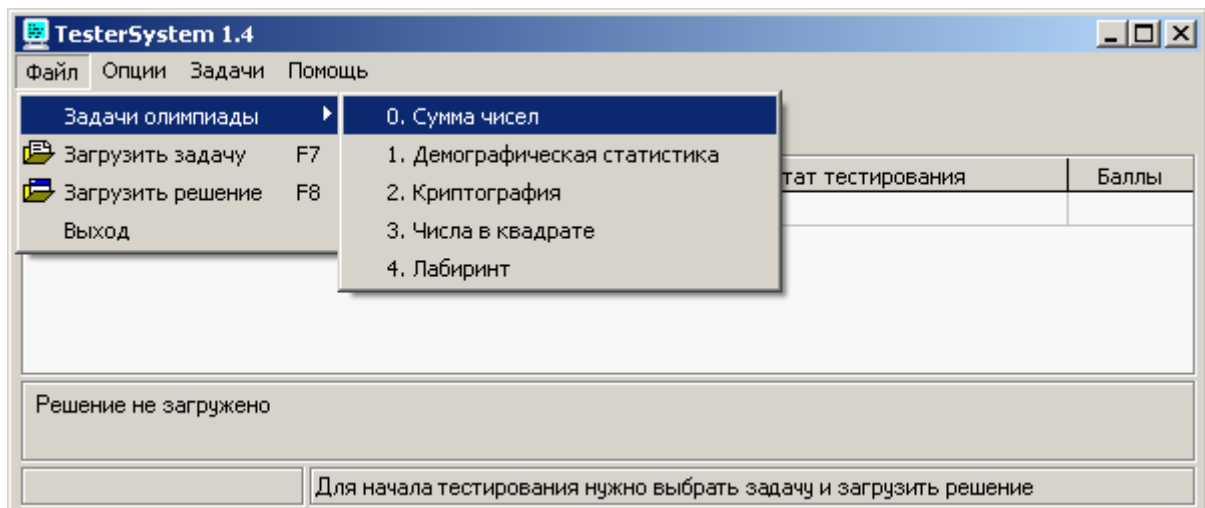
Никаких `ReadLn`; и другого кода, который предполагает действие пользователя для завершения программы, быть не должно. В противном случае такое решение тестирование не пройдет из-за выхода лимита времени, отводимого под решение каждой задачи.

Интерфейс тестирующей программой

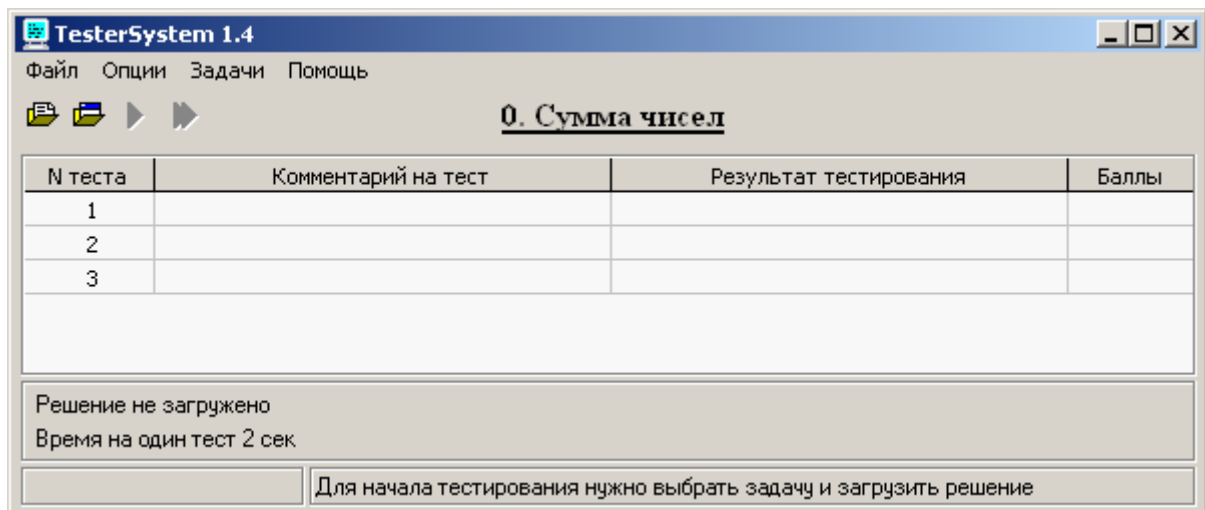
При запуске тестирующей программы она будет иметь следующий вид:




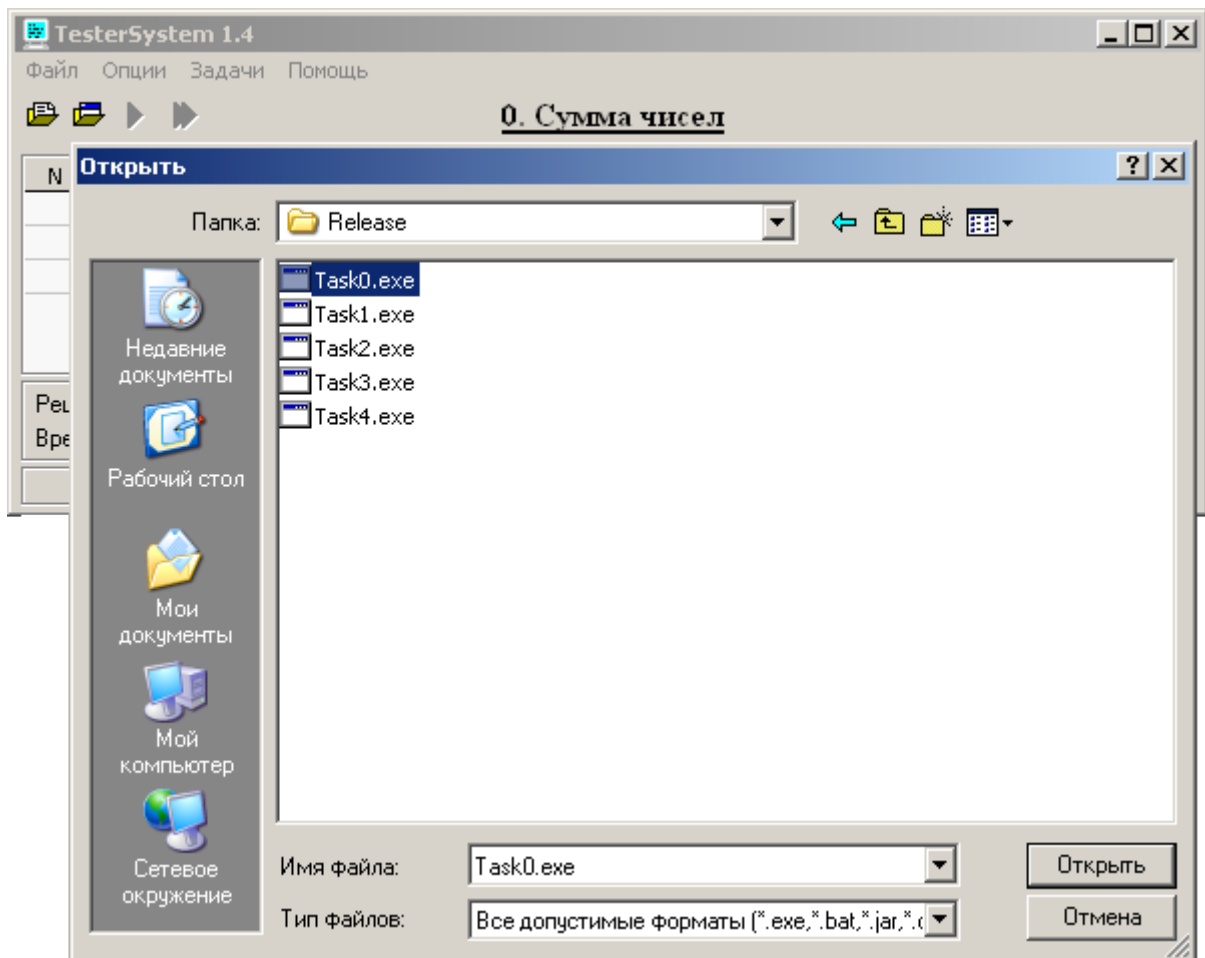
Прежде чем приступить к тестированию решения необходимо выбрать конкретную задачу (в терминах программы «загрузить задачу»). Для этого необходимо воспользоваться кнопкой  и выбрать папку с задачей. Или же, если корректно сконфигурирована в файле TS.ini программа настроена, выбрать интересующую задачу из меню:




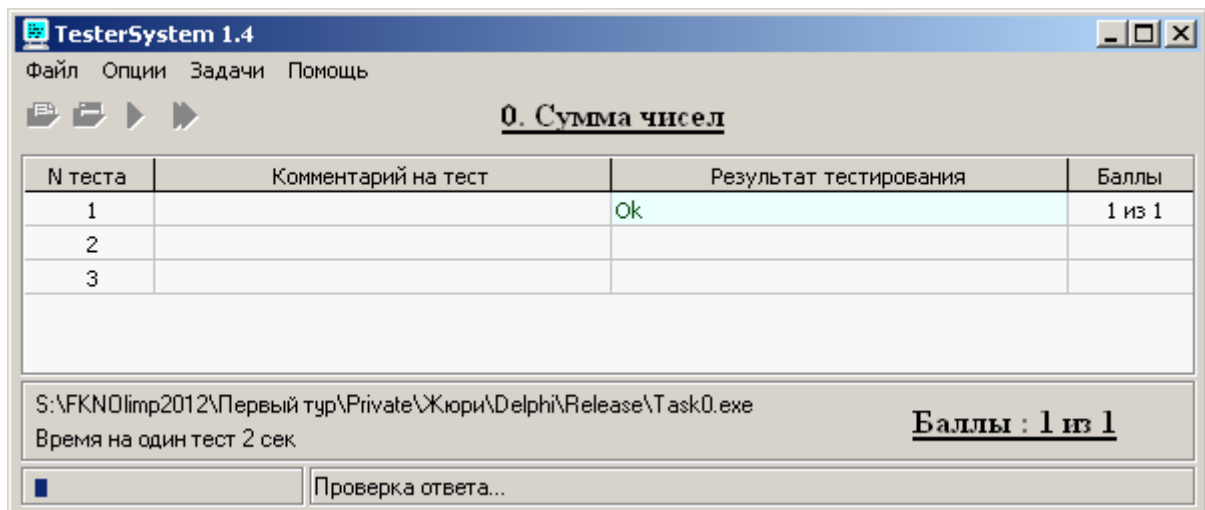
После выбора задачи программа будет выглядеть следующим образом:



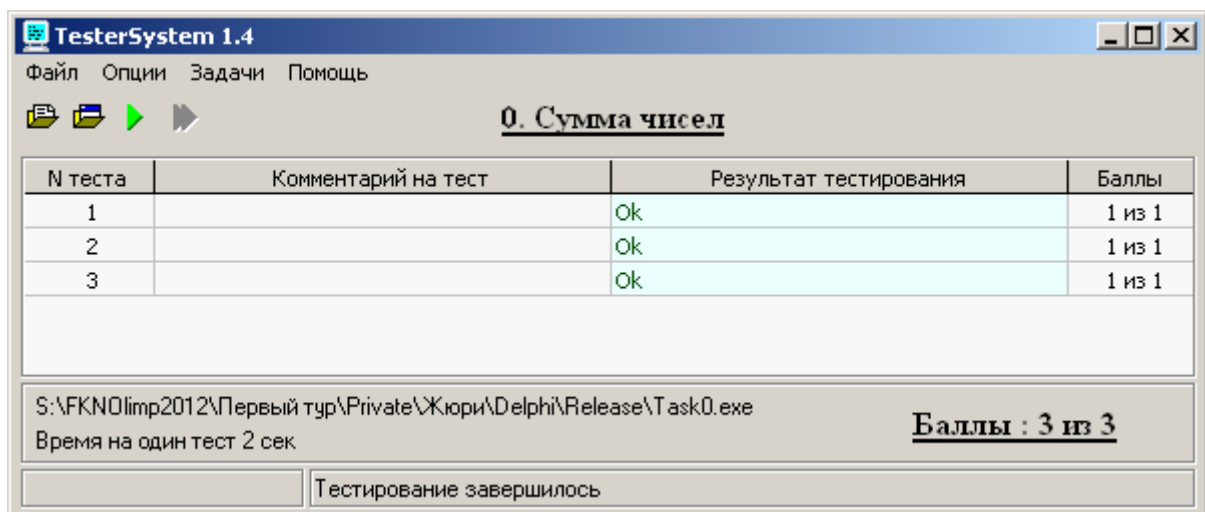
Далее необходимо выбрать тестируемое решение (в терминах программы «загрузить решение»). Для этого необходимо воспользоваться кнопкой  и выбрать исполняемый (*.exe, *.jar и т.д.) файл решения (скомпилированное решение участника):



После этого станет доступна кнопка  («тестирование решения»). Нажатие этой кнопки приведет к запуску тестирования решения с помощью тестов к данной задаче:



По завершению тестирования программа будет выглядеть следующим образом (для решения, проходящего все тесты):



В случае, если решение не пройдет какие-то тесты, в столбце «Результат тестирования» будет отображена соответствующая информация.

Далее можно перетестировать исправленное решение или же приступить к тестированию другой задачи (выбрав другую задачу и другое решение).

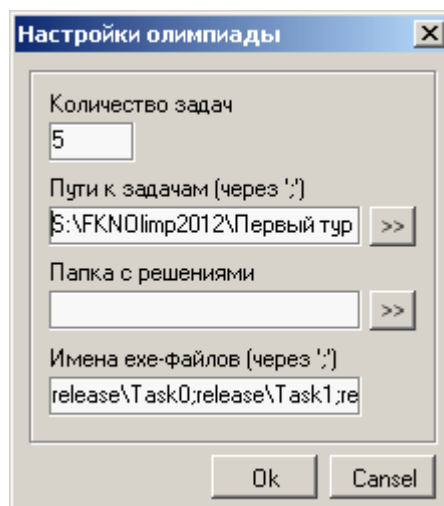
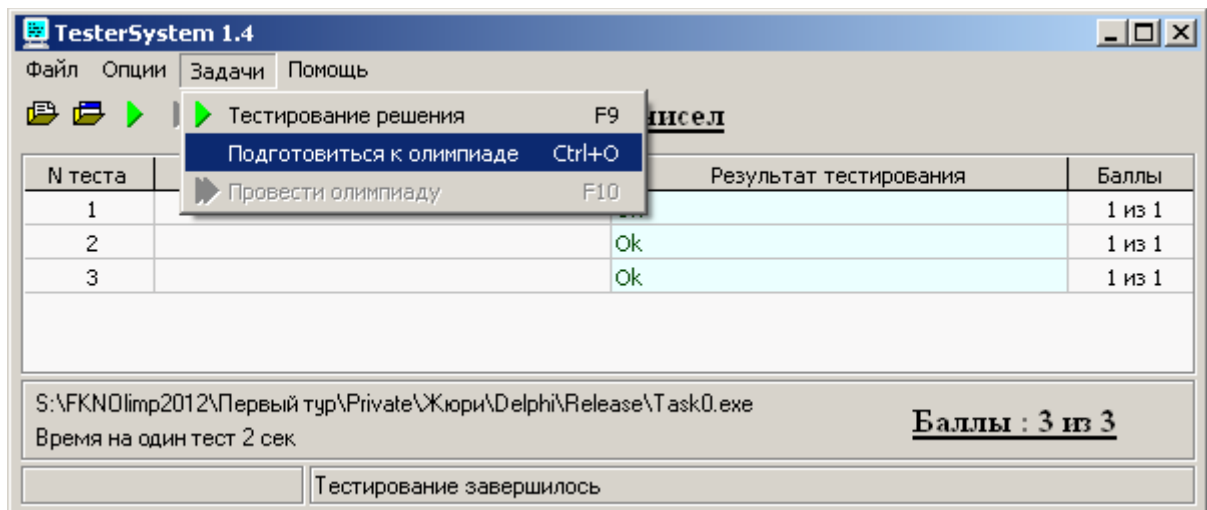
Пакетное тестирование

Данная тестирующая программа позволяет тестировать все решения участников в пакетном режиме. Для этого все решения участников, оформленные и размещенные в надлежащем виде (см. Правила проведения олимпиады по программированию) должны находиться в одной директории, например, так:

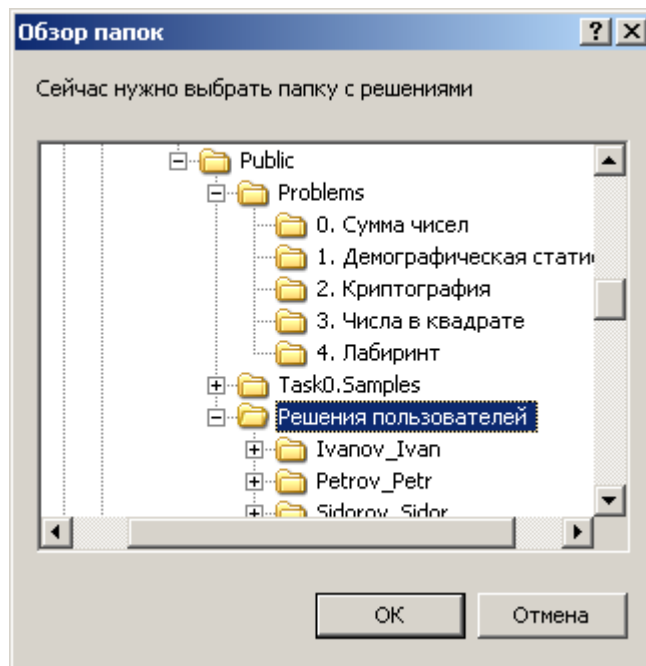
```
Решения участников
  Ivanov_Ivan
    Index.txt
    Release
      Task1.exe
      Task2.exe
      Task3.exe
      Task4.exe
    Source
      Task1
```


```
..._исходные_тексты_задачи_1_...
Task2
..._исходные_тексты_задачи_2_...
Task3
..._исходные_тексты_задачи_3_...
Task4
..._исходные_тексты_задачи_4_...
Petrov_Petr
..._решения_участника_согласно_правилам_...
Sidorov_Sidor
..._решения_участника_согласно_правилам_...
..._другие_участники_...
```

Для пакетного тестирования необходимо указать программе тестирования размещение задач и решений пользователей в отдельном диалоге (в терминах программы «подготовиться к тестированию»):

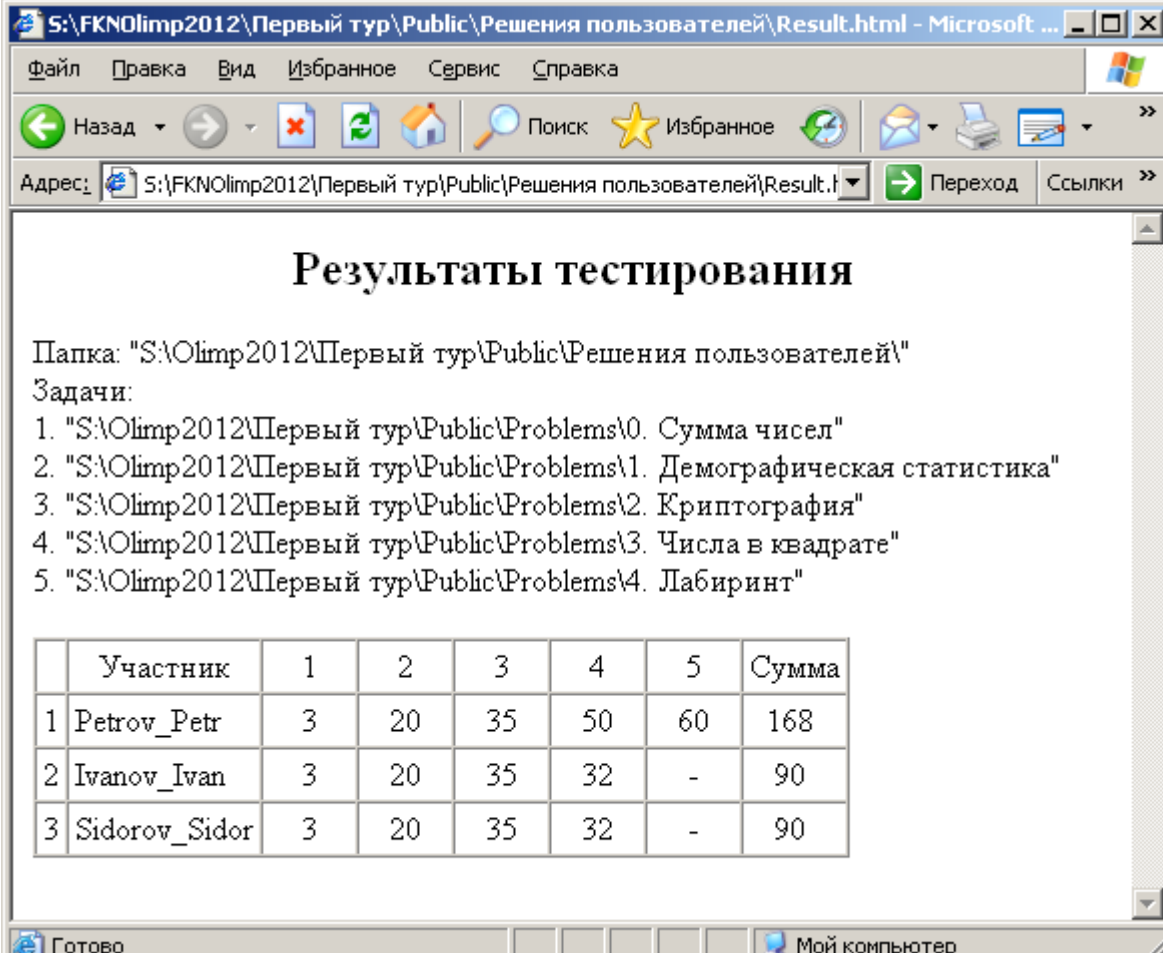


В данном диалоге необходимо указать количество задач, пути к задачам через точку с запятой, директорию с решениями участников и размещение тестируемых исполняемых файлов (без расширения) относительно директории пользователя. Если программа тестирования сконфигурирована в TS.ini, то необходимо будет выбрать только «Папку с решениями»:



После того, как подготовка к пакетному тестированию будет завершена, станет доступна кнопка , запускающая пакетное тестирование.

После проведения пакетного тестирования результаты тестирования будут сохранены в файле Results.html, например так:



Результаты тестирования

Папка: "S:\Olimp2012\Первый тур\Public\Решения пользователей"

Задачи:

- "S:\Olimp2012\Первый тур\Public\Problems\0. Сумма чисел"
- "S:\Olimp2012\Первый тур\Public\Problems\1. Демографическая статистика"
- "S:\Olimp2012\Первый тур\Public\Problems\2. Криптография"
- "S:\Olimp2012\Первый тур\Public\Problems\3. Числа в квадрате"
- "S:\Olimp2012\Первый тур\Public\Problems\4. Лабиринт"

	Участник	1	2	3	4	5	Сумма
1	Petrov_Petr	3	20	35	50	60	168
2	Ivanov_Ivan	3	20	35	32	-	90
3	Sidorov_Sidor	3	20	35	32	-	90

Готово

Для приведенного примера осталось вычесть результаты первой тестовой задачи из суммы баллов каждого участника (например в Excel) можно объявлять результаты.

Замечание:

Ошибочные решения участников, которые приводят к открытию дополнительных окон отладки, отправки информации об ошибке разработчику и т.п., следует исключить из пакетного тестирования во избежание влияния на результат тестирования других решений и протестировать отдельно.

Тестирование решений, написанных на Java

Тестирующая программа TS.exe позволяет тестировать решения, написанные на разных языках, если такое решение можно каким-либо образом запустить из командной строки. Правила запуска файлов, не являющихся исполняемыми файлами Windows (*.exe), задаются в конфигурационном файле программы TS.ini.

Скомпилированное Java-решение не является исполняемым файлом Windows. Скомпилированные классы имеют расширение *.class и запускаются с помощью интерпретатора Java байт-кода (программа java.exe). Правила для запуска *.class файлов в TS.ini по умолчанию прописаны, поэтому для тестирования Java-решений необходимо лишь наличие установленного JDK или JRE соответствующей версии на компьютере, на котором производится тестирование (команда java должна быть доступна из командной строки).

Если решение участника оформлено в виде одного класса, то тестировать можно скомпилированную версию данного класса, например Task0.class (следует помнить, что имя класса должно совпадать с именем файла). Если в решении используется несколько классов и при компиляции соответственно будет получено несколько *.class файлов, то для тестирования решения *.class файлы необходимо упаковать в исполняемый jar-пакет (executable jar file), например, Task0.jar, и при тестировании выбирать полученный *.jar файл.