

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ ЛУГАНСКОЙ НАРОДНОЙ РЕСПУБЛИКИ
ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ ЛУГАНСКОЙ НАРОДНОЙ РЕСПУБЛИКИ
«ДОНБАССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
ИНСТИТУТ»
(ГОУ ВО ЛНР «ДонГТИ»)

Факультет автоматизации и электротехнических систем
Кафедра Специализированных компьютерных систем



УТВЕРЖДАЮ
Ректор

Д.А. Вишневский

ПОЛОЖЕНИЕ
о проведении олимпиады по программированию
(в рамках I этапа Республиканской студенческой олимпиады по программированию)

Рассмотрено и одобрено
на заседании кафедры СКС
Протокол № 7 от 14.03.2022

Врио зав. кафедрой СКС  проф. Е.Е. Бизянов

Алчевск, 2022

1. ОБЩИЕ ПОЛОЖЕНИЯ

1.1 Цель олимпиады

Целью проведения олимпиады по программированию (в рамках I этапа Республиканской студенческой олимпиады по программированию) (далее «олимпиада») является создание условий для повышения уровня знаний и оценки подготовки студентов в области прикладного программирования.

1.2 Организаторы олимпиады

Организатором проведения олимпиады является кафедра «Специализированные компьютерные системы» Государственного образовательного учреждения Луганской Народной Республики «Донбасской государственный технический институт» в лице заведующего кафедрой.

Составы оргкомитета Олимпиады, методической и апелляционной комиссий и жюри по предметам утверждаются ректором Института.

Одновременное членство лиц в методической комиссии и жюри Олимпиады не допускается.

1.2.1. Оргкомитет Олимпиады:

обеспечивает непосредственное проведение Олимпиады;

формирует и представляет на утверждение состав методической комиссии и жюри по предметам, осуществляет координацию их деятельности;

определяет порядок, график и места проведения Олимпиады;

распространяет информацию о проведении Олимпиады;

утверждает отчеты методической комиссии и жюри по предметам;

разрабатывает систему регистрации, учёта участников Олимпиады и сданных работ, изготавливает необходимые для этого материалы;

организует и контролирует прием олимпиадных работ и заполненных регистрационных карточек (анкет участников), процесс выдачи работ на проверку членам жюри, приём проверенных работ и контролирует их сохранность;

организует компьютерный набор сведений из анкет участников с целью последующей автоматизированной обработки результатов Олимпиады, результатов проверки работ, контролирует полноту и достоверность учёта результатов проверки;

осуществляет оформление и учет дипломов победителей и призеров Олимпиады;

организует хранение невыданных дипломов и их выдачу в рабочем порядке по мере обращения участников Олимпиады;

рассматривает конфликтные ситуации;

утверждает абсолютного победителя Олимпиады;

готовит проект приказа, утверждающего списки победителей и призеров Олимпиады;

награждает победителей и призеров Олимпиады.

1.2.2. Методическая комиссия Олимпиады по предмету:

организует подготовку заданий Олимпиады на высоком научном и методическом уровне;

представляет подготовленные задания в оргкомитет не позднее 15 дней до даты проведения этапов Олимпиады, совместно с оргкомитетом участвует в подготовке материалов Олимпиады, проверяет и утверждает окончательный вариант макета заданий;

разрабатывает критерии и методики оценки выполненных заданий Олимпиады;

вносит предложения в оргкомитет по вопросам, связанным с совершенствованием организации проведения и методического обеспечения Олимпиады;

совместно с жюри участвует в подведении итогов Олимпиады по предмету;

совместно с оргкомитетом участвует в рассмотрении конфликтных ситуаций;

публикует решения олимпиадных заданий.

1.2.3. Жюри Олимпиады по предмету:

проводит проверку работ участников Олимпиады, оценивает результаты выполнения заданий;

определяет совместно с оргкомитетом победителей и призеров Олимпиады, готовит предложения по их награждению;

проводит анализ выполнения заданий участниками Олимпиады;

составляет протокол результатов Олимпиады по предметам;

осуществляет иные функции в соответствии с Положением об Олимпиаде.

1.2.4. Апелляционная комиссия:

состоит из членов оргкомитета, методической комиссии, жюри, а также экспертов, не являющихся членами вышеназванных органов, обладающих необходимой квалификацией и знаниями в соответствующей области;

рассматривает апелляцию о несогласии с выставленными баллами, поданную участником Олимпиады;

принимает решение об отклонении и сохранении выставленных баллов либо об удовлетворении апелляции и выставлении других баллов, причем по результатам рассмотрения апелляции количество выставленных баллов может быть изменено как в сторону увеличения, так и в сторону уменьшения;

передает результаты апелляции участника Олимпиады в Оргкомитет;

осуществляет иные функции в соответствии с Положением об Олимпиаде.

1.3 Участники олимпиады

1.3.1 Образовательные учреждения высшего образования.

Участниками олимпиады являются команды, представляющие образовательные учреждения высшего образования. Каждая команда состоит из 1–3 участников. У команды должен быть тренер, через которого осуществляется решение всех вопросов относительно организации и принятия участия в олимпиаде.

Команды участвуют в олимпиаде дистанционно на базе кафедры «Специализированные компьютерные системы» ГОУ ВО ЛНР «ДонГТИ».

Для участия в олимпиаде команды до 22 апреля 2022г. (включительно) подают заявку на участие в олимпиаде (приложение А).

1.3.2 Общеобразовательные учреждения

Представители общеобразовательных учреждений принимают участие в олимпиаде в отдельной категории.

Представители общеобразовательных учреждений участвуют в олимпиаде дистанционно с использованием системы дистанционного обучения ГОУ ВО ЛНР «ДонГТИ».

Для участия в олимпиаде до 22 апреля 2022г. (включительно) участники подают заявку на участие в олимпиаде (приложение Б). Заявка подписывается

руководителем образовательного учреждения или его заместителем. Электронная копия заявки пересылается на указанную в Приложениях почту.

2. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Проверка решений олимпиадных задач производится членами жюри.

Исходные тексты программ (текст) и результаты их выполнения (полученные значения и копия экрана) загружаются на страницу «Олимпиада по программированию 2022» по ссылкам, находящимся внизу под каждым заданием.

Возможно использование компиляторов:

- GNU C;
- GNU C++;
- Free Pascal;
- Java;
- Mono Visual Basic.

Участники при оформлении заявки должны указать, какие из компиляторов, перечисленных выше, они будут использовать.

3. ПОРЯДОК ПРОВЕДЕНИЯ

3.1. Правила проведения

Олимпиада проводится в один тур по правилам ACM ICPC, которые предусматривают следующее:

- решение считается принятым, если оно проходит все тесты;
- принятое решение получает штрафной балл, равный количеству минут, прошедших от начала турнира;
- участники ранжируются в порядке уменьшения числа принятых задач, а при равенстве задач в порядке увеличения штрафного балла.

Продолжительность тура составляет 4 астрономических часа. Во время тура все участники решают один и тот же набор из 7 задач. Каждая задача имеет название, формулировку и примеры входных/выходных данных.

При решении задач запрещено:

1. Создавать новые процессы;
2. Открывать сетевые соединения;
3. Выполнять чтение и запись векторов прерываний;
4. Обращаться к системным функциям операционной системы;
5. Создавать диалоговые окна (как графические, так и текстовые);
6. Работать с любыми файлами и каталогами, не оговоренными в условии задачи;
7. Использовать любые приемы и методы, способные помешать процессу проведения олимпиады.

В случае нарушения правил, судейская коллегия имеет право отстранить участника от участия в олимпиаде.

Участники имеют право общаться с представителями группы технической поддержки и судейской коллегии.

Во время олимпиады участники имеют возможность задавать вопросы, связанные с уточнением условий задач. Для этого участник может задавать вопрос через web-интерфейс системы. Судейская коллегия рассматривает вопрос и передает ответ. Вопросы должны формулироваться так, чтобы на них можно было дать ответ «да» или «нет». Если вопрос не относится к условию задачи, его формулировка непонятна, ответ на него очевиден, ответ на него может содержать «подсказки» и т. п., то судейская коллегия может оставить такой вопрос без ответа.

Во время тура имеется возможность просматривать текущие результаты на своих компьютерах через web-интерфейс системы. Таблица результатов «замораживается» за 1 астрономический час до окончания соревнования: у команд (участников) есть возможность видеть вердикты по своим послылкам и наличие посылок по задачам других команд (участников) без указания, была эта попытка успешной или нет.

После окончания тура попытки не принимаются. В случае возникновения непредвиденных обстоятельств во время проведения олимпиады судейская коллегия имеет право на продление времени проведения тура.

4. РЕШЕНИЕ ЗАДАЧ

4.1 Оформление задач

Во время тура участники решают предложенные задачи. Решенная задача представляется в виде файла в 8-битной кодировке (cp1251, cp866, koï8-г, utf8) с исходным текстом программы на одном из языков программирования, указанных в п. 2.1.

Решение не может содержать иных частей, таких как модули, пакеты (кроме стандартных) оформленных отдельно от основной программы. Все решения должны быть оформлены в виде консольных приложений.

В конце работы программа должна закрыть все открытые файлы (если таковые имеются) и завершиться с кодом возврата 0. Во всех языках программирования кроме Си и C++ нулевой код завершения программы вырабатывается автоматически при её нормальном завершении. В языках Си и C++ в конце функции main необходимо использовать оператор return 0.

Программа должна считывать входные данные со стандартного потока ввода (считывать с клавиатуры), результат работы должен выводиться на стандартный поток вывода (экран), если иное не оговорено условиями задачи. Во всех языках программирования, перечисленных в п. 2, предусмотрены функции вывода данных на стандартный поток вывода и ввода данных со стандартного потока ввода.

В качестве примера оформления решений рассмотрено решение задачи нахождения суммы двух чисел (далее «A+B»).

4.1.1. Оформление решений на языке Си

В программах на языке Си для чтения данных следует использовать функции scanf, getchar и т. д., а для вывода результата — функции printf, putchar и т. д. Вместо типа данных __int64, поддерживаемого компилятором Microsoft C/C++, необходимо использовать тип данных long long.

В систему для проверки необходимо отправлять файл *.c. Ниже приведен пример решения задачи A+B:

```
#include <stdio.h>
```

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

    scanf("%d%d", &a, &b);
    printf("%d\n", a + b);

    return 0;
}
```

4.1.2. Оформление решений на языке C++

В программах на языке C++ можно использовать как функции ввода-вывода языка Си, так и операции чтения из `cin` и записи в `cout`. Вместо типа данных `__int64`, поддерживаемого компилятором Microsoft C/C++, необходимо использовать тип данных `long long`.

В систему для проверки необходимо отправлять файл `*.cpp`. Ниже приведен пример решения задачи A+B:

```
#include <iostream>

using namespace std;

int main()
{
    int a, b;

    cin >> a;
    cin >> b;
    cout << a + b;

    return 0;
}
```

Во время выполнения допускаются исключения, возникающие внутри блока `try..catch`. Они будут проигнорированы системой. Исключения, возникающие вне блока `try...catch` приведут к Runtime Error.

4.1.3. Оформление решений на языке Free Pascal

В программах на языке Pascal для чтения следует использовать процедуры `read` или `readln` без файлового параметра, а для записи — процедуры `write` или `writeln` без файлового параметра.

В систему для проверки необходимо отправлять файл *.pas. Ниже приведены примеры решения задачи A+B:

```
program sum;
var a, b, c : longint;

begin
    readln(a);
    readln(b);

    c := a + b;
    writeln(c);
end.
```

Во время выполнения допускаются исключения, возникающие внутри блока try..except. Они будут проигнорированы системой. Исключения, возникающие вне блока try...except приведут к Runtime Error.

4.1.4. Оформление решений на языке Java

В программах на языке Java для чтения следует использовать методы работы с потоком System.in, а результат выводить в поток System.out. Для чтения данных можно использовать класс Scanner, но рекомендуется создавать StreamTokenizer.

На сервер для проверки необходимо отправлять файл *. java имя файла должно совпадать с именем класса, содержащегося в файле, для рассмотренного примера имя файла должно быть Example.java. Ниже приведен пример решения задачи A+B:

```
import java.io.*;
import java.util.Scanner;

public final class Example
{
    public static void main(String args[]) throws Exception
    {
        Scanner s = new Scanner(System.in);
        int a, b;
        a = s.nextInt();
        b = s.nextInt();
        System.out.println(a + b);
    }
}
```

4.1.5. Оформление решений на языке VB.NET (компилятор mono Visual Basic)

В программах на языке VB.NET для чтения следует использовать методы работы с объектом Console: ReadLine, WriteLine. Следует учитывать, что система использует компилятор mono Visual Basic, который не в полной мере может поддерживать последние возможности языка Visual Basic .NET.

В систему для проверки необходимо отправлять файл *.vb. Ниже приведен пример решения задачи A+B:

```
Module ab
  Sub Main()
    Dim a, b, c as Integer
    a = Console.ReadLine()
    b = Console.ReadLine()
    c = a + b
    Console.WriteLine(c)
  End Sub
End Module
```

4.2 Особенности языков и сред программирования

Предложенный в п.2 набор компиляторов работает под управлением операционной системы Linux, поэтому необходимо учитывать ряд особенностей приведенных ниже.

4.2.1 Компиляторы gcc и g++

В среде Linux в качестве разделителей каталогов в пути используется только символ «/», в отличие от Windows, где допускается как «/», так и «\». Поэтому в директивах #include должен использоваться только символ «/».

Неправильно:

```
#include <sys\types.h>
```

Правильно:

```
#include <sys/types.h>
```

4.2.2 Языки Си, C++

В среде Visual C++ для определения 64-битных целых типов используется тип __int64. Visual C++, начиная с версии 2005, поддерживает и стандартный 64-битный

тип long long.

Обратите внимание, что компиляторы GCC, работающие на Windows (в среде Cygwin или MinGW), поддерживают только стандартный 64-битный тип long long.

4.2.3 Язык C++

Потоковый ввод и вывод языка C++ примерно в 10–20 раз медленнее функций printf и scanf. Если требуется ввести или вывести большое количество данных, используйте функции ввода-вывода языка Си.

4.3 Проверка решений

Проверка представленных решений проводится после окончания тура. Участники загружают решения с помощью web-интерфейса системы. После проверки решения жюри выносит вердикт и участнику сообщается, зачтено решение или нет.

Исходный текст программы компилируется указанным при отправке компилятором. Если при компиляции решения возникают ошибки, проверка решения на этом завершается со статусом «Ошибка компиляции» («Compilation error»). Попытки сдачи программ, приведшие к ошибке компиляции, не учитываются при начислении штрафных баллов, но засчитываются как лишние использованные попытки.

Решение проверяется путем запуска на наборе тестов, который недоступен участникам и является одинаковым для всех. Все входные и выходные данные тестов являются корректными и удовлетворяющими всем ограничениям, указанным в условии задачи.

Во время работы тестируемой программы контролируется ограничение процессорного времени, ограничение реального (астрономического) времени и ограничение размера используемой памяти на каждом тесте.

Программа считается прошедшей тест, если при запуске на этом тесте программа уложилась в ограничения по времени и по памяти, завершилась нормально (с кодом завершения 0) и выдала правильный результат. В остальных случаях программа считается не прошедшей тест. Решение засчитывается в том случае, если оно выдает верные ответы на все тесты.

Все представленные участниками программы должны в точности соблюдать форматы ввода-вывода, описанные в условии каждой задачи.

5. ПОРЯДОК ПОДАЧИ АПЕЛЛЯЦИЙ

Апелляция подается участником в апелляционную комиссию на протяжении одного часа после получения результатов тура соревнований. Поводом для апелляций может быть технический сбой системы, связанный с неправильной регистрацией и оценкой решений.

Апелляция рассматривается членами апелляционной комиссии сразу после ее подачи. Результатом апелляции может быть пересмотр оценок участника, который подала апелляцию, или ее отклонение.

Решение апелляционной комиссии является окончательным и не подлежит пересмотру.

6. ПОДВЕДЕНИЕ ИТОГОВ СОРЕВНОВАНИЯ

На основании рейтингов участников и апелляций члены судейской коллегии определяют победителей олимпиады. Победителем олимпиады является участник, набравший наибольший рейтинг.

Все участники олимпиады получают сертификат участника. Участники-победители олимпиады, занявшие 1–3 места, в своих категориях, получают дипломы.

Результаты олимпиады оглашаются после завершения тура и рассмотрения апелляций.

Судейская коллегия обладает исключительным правом определения правильности присланных решений, определения победителей и дисквалификации команд.

ПРИЛОЖЕНИЕ А

ЗАЯВКА НА УЧАСТИЕ В ОЛИМПИАДЕ

от образовательных учреждений высшего образования

в Оргкомитет олимпиады по программированию (в рамках I этапа Республиканской
студенческой олимпиады по программированию)

а) Информация об участнике:

Название участника: _____

ФИО (полностью): _____

Дата рождения: _____

Факультет: _____

Специальность: _____

Курс обучения: _____

E-mail: _____

Контактный телефон: _____

Полное название учебного заведения: _____

Сокращенное название учебного заведения: _____

Город: _____

Адрес учебного заведения: _____

Языки программирования и среды разработки: _____

б) Информация о тренере:

ФИО (полностью): _____

Дата рождения: _____

Факультет: _____

Должность: _____

E-mail: _____

Контактный телефон: _____

Примечание

Олимпиада будет проведена на базе кафедры «Специализированные компьютерные системы» ГОУ ВО ЛНР «ДонГТИ» 26, 27 апреля 2022г.

График проведения: 26 апреля – тестирование входа в систему и работы с ней: 14-30;

27 апреля – основной тур: 9-00.

Заявки необходимо отправлять на электронную почту:

bee@dstu.education

Телефон для справок: 072-134-14-38

ПРИЛОЖЕНИЕ Б

ЗАЯВКА НА УЧАСТИЕ В ОЛИМПИАДЕ

от общеобразовательных учреждений в Оргкомитет олимпиады по
программированию (в рамках I этапа Республиканской студенческой
олимпиады по программированию)

а) Информация об участнике:

Название участника: _____

ФИО (полностью): _____

Дата рождения: _____

Факультет: _____

Специальность: _____

Курс обучения: _____

E-mail: _____

Контактный телефон: _____

Полное название учебного заведения: _____

Сокращенное название учебного заведения: _____

Город: _____

Адрес учебного заведения: _____

Языки программирования и среды разработки: _____

б) Информация о тренере:

ФИО (полностью): _____

Дата рождения: _____

Факультет: _____

Должность: _____

E-mail: _____

Контактный телефон: _____

Примечание

Олимпиада будет проведена на базе кафедры «Специализированные компьютерные системы» ГОУ ВО ЛНР «ДонГТИ» 26, 27 апреля 2022г.

График проведения: 26 апреля – тестирование входа в систему и работы с ней: 14-30;

27 апреля – основной тур: 9-00.

Заявки необходимо отправлять на электронную почту:

bee@dstu.education

Телефон для справок: 072-134-14-38