

Государственное образовательное учреждение
высшего образования
Луганской Народной Республики
«Донбасский государственный технический институт»

Факультет автоматизированных и электротехнических систем

Кафедра специализированных компьютерных систем



УТВЕРЖДАЮ

Первый проректор

_____ А.В. Кунченко

ПОЛОЖЕНИЕ

о проведении олимпиады по программированию

(в рамках I этапа Республиканской студенческой олимпиады по
программированию)

Рассмотрено и одобрено

на заседании кафедры СКС

Протокол № 4 от 24.10.2022 г.

Врио зав.кафедрой СКС _____

Е.Е. Бизянов

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» Донбасского государственного технического института 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

Участниками олимпиады являются команды, представляющие образовательные учреждения высшего образования. Каждая команда состоит из 2–3 участников. У команды должен быть тренер, через которого осуществляется решение всех вопросов относительно организации и принятия участия в олимпиаде.

Команды участвуют в олимпиаде очно на базе кафедры «Специализированные компьютерные системы» Донбасского государственного технического института.

Для участия в олимпиаде команды до 21 февраля 2022г. (включительно) подают заявку на участие в олимпиаде (приложение А).

2. АППАРАТНОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

2.1. Проверяющая система

Проверка решений олимпиадных задач происходит с помощью проверяющей системы e-judge версии 3.7.6 (далее «система»).

Доступные компиляторы:

- GNU C (7.2.1);

- GNU C++ (7.2.1);
- Free Pascal (3.0.2);
- Java (JDK 1.8.0);
- Mono Visual Basic.

2.2. Рабочие места

Во время проведения олимпиады команде (участнику) предоставляется подключенный к локальной сети института IBM PC-совместимый компьютер в следующей конфигурации:

- процессор Pentium или совместимый;
- ОЗУ не менее 512 Мб;
- операционная система семейства Windows или Linux;
- сопутствующее программное обеспечение (компиляторы, среды разработки).

Участники при оформлении заявки должны указать, какие среды разработки им потребуются для компиляторов, перечисленных в п. 2.1. В противном случае гарантируется только наличие сред разработки, устанавливаемых по умолчанию совместно с компиляторами, перечисленными в п. 2.1.

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, плюс 20 штрафных минут за каждую неуспешную попытку по данной задаче, которые предшествовали первой успешной попытке;
- 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 команды (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 команду (участника) от участия в олимпиаде.

Во время олимпиады члены команды имеют право общаться между собой. Запрещено обсуждать идеи и обмениваться решениями задач вне команды. Команды (участники) имеют право общаться с представителями группы технической поддержки и судейской коллегии.

Во время олимпиады команды (участники) имеют возможность задавать вопросы, связанные с уточнением условий задач. Для этого команда (участник) может задавать вопрос через web-интерфейс системы. Судейская коллегия рассматривает вопрос и передает команде 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 всех команд (участников) 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 посылкам 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.

3.2. Прочие условия

Участники могут приносить с собой чистые листы бумаги, и инструменты для письма. Запрещается использовать любые (кроме предоставленных жюри) книги, распечатки, электронные средства хранения и передачи информации:

- личные компьютеры/ноутбуки;
- планшетные компьютеры;
- flash-накопители, CD/DVD диски;
- калькуляторы;
- мобильные телефоны;
- и др.

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-r, 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 С++ нулевой код завершения программы вырабатывается автоматически при её нормальном завершении. В языках Си и С++ 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. 2.1. 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 MicrosoftC/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 MicrosoftC/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 FreePascal

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 main.java. Ниже приведен пример решения задачи A+B:

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

public final class main
{
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 monoVisualBasic, который не в полной мере может поддерживать последние возможности языка VisualBasic .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)
EndSub
EndModule
```

4.2 Особенности языков и сред программирования

Система и предоставляемый набор компиляторов работают 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.

Подробнее об особенностях, которые необходимо учитывать при оформлении решений, можно ознакомиться в справочной документации системы.

4.3 Проверка решений

Проверка представленных решений проводится во время тура. Участники посылают решения в систему с помощью web-интерфейса системы. После проверки решения система 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»). В случае ошибки компиляции участник может просмотреть полный вывод компилятора, чтобы установить и устранить причину ошибки. Попытки сдачи, приведшие к ошибке компиляции, не учитываются при вычислении штрафных баллов, но засчитываются как 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.

Тестирование производится автоматически, поэтому программы должны в точности соблюдать форматы ввода-вывода, описанные в условии каждой задачи.

Все тесты выполняются в одном и том же порядке. Номер ошибочного теста, указываемый в результатах проверки — номер первого теста, на котором задача не прошла проверку. Например, если система сообщила: «Time-limit exceeded: Test 9» («Превышен предел времени на тесте № 9»), это означает:

- тесты с 1 по 8 прошли проверку;
- на более сложном, чем предыдущие восемь, девятом тесте программа превысила предел времени.

Если получено сообщение «Presentation error» («Неправильный формат вывода»), то это означает, что формат вывода результата программы отличается от требуемого.

Сообщение «Run-time error» означает ошибку времени выполнения (аварийное завершение программы).

Сообщение «Wrong answer» означает неправильный ответ.

Подробнее о правилах работы в системе можно ознакомиться в справочной документации системы.

5. ПОРЯДОК ПОДАЧИ АПЕЛЛЯЦИЙ

Апелляция подается на протяжении одного часа после окончания тура соревнований представителем команды в апелляционную комиссию. Поводом для апелляции 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 команды, которая подала апелляцию, или ее отклонение.

Решение апелляционной комиссии является окончательным и не подлежит пересмотру.

6. ПОДВЕДЕНИЕ ИТОГОВ СОРЕВНОВАНИЯ

На основании рейтингов команд (участников) и апелляций члены судейской коллегии определяют победителей олимпиады. Победителем олимпиады является команда (участник), набравшие наибольший рейтинг.

Все участники олимпиады получают сертификат участника. Команды (участники) победители олимпиады, занявшие 1–3 места, в своих категориях, получают дипломы.

Результаты олимпиады оглашаются в день проведения после завершения тура и рассмотрения апелляций. Так же на протяжении 3-х рабочих дней после завершения тура олимпиады информация об итогах размещается на сайте ГОУ ВО ЛНР «ДонГТИ»

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 о команде:

Название команды: _____

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: _____

с) Информация о каждом члене команды:

ФИО (полностью): _____

Дата рождения: _____

Факультет: _____

Специальность: _____

Курс обучения: _____

E-mail: _____

Контактный телефон: _____

ФИО (полностью): _____
Дата рождения: _____
Факультет: _____
Специальность: _____
Курс обучения: _____
E-mail: _____

Контактный телефон: _____

ФИО (полностью): _____
Дата рождения: _____
Факультет: _____
Специальность: _____
Курс обучения: _____
E-mail: _____
Контактный телефон: _____

Примечание

Олимпиада будет проведена на базе кафедры «Специализированные компьютерные системы» ГОУ ВО ЛНР «ДонГТИ» 07 декабря 2022г. по адресу: ул. Чапаева, 49-А, IV учебный корпус, ауд.4207.

График проведения: 06 декабря 2022 г. – инструктаж: 16-00;
07 декабря 2022 г. – основной тур: 13-00.

Заявки необходимо отправлять на электронную почту:

kafedra.scs@yandex.ru

Телефон для справок: 072-134-14-38