

Олимпиада «ИТ-Планета 2013/14»
Конкурс по программированию: Java

Инструкция по участию во втором отборочном этапе

1. Дата и время проведения соревнований

Дата проведения соревнований: **5 марта** 2014 г.

Время на ознакомление с системой: с **10:00** до 10:30 по московскому времени.

Время на решение задач: с **10:30** до 14:30 по московскому времени.

2. Доступ к серверу

Доступ к серверу осуществляется из личного кабинета на сайте олимпиады, раздела «Мои тесты».

Название конкурса	Тесты
Конкурс по программированию: Java	Демо-тест ТЕСТ ТЕСТ 2-го тура

3. Требования к программному обеспечению и оформлению программ

На тестирующей системе установлена Oracle JDK 7 u51, рекомендуем участникам также использовать самую свежую версию JDK, которую можно скачать по [ссылке](#). Используемая версия Java - Java SE 7u51.

3. Требования к оформлению решений

Некоторые решения не требуют наличия метода main(), такие задачи проверяются Unit-тестами. Решение такой задачи должно имплементировать заданный в условиях интерфейс. Класс объявляется вне пакета. Если в задаче явно не указан интерфейс, то файл, который Вы посылаете на проверку, должен содержать класс с методом main(), который объявляется в безымянном пакете (вне пакета).

Все решения должны состоять из одного файла с исходным текстом. Проекты, состоящие более чем из одного исходного файла не поддерживаются.

В зависимости от условия задачи программа должна считывать входные данные со стандартного потока ввода (часто в этом случае пишут «считывать с клавиатуры») или из заданного файла. В зависимости от условия задачи результат работы должен выводиться либо на стандартный поток вывода (часто называемый «экраном»), либо в заданный файл. Никакие другие файлы использовать нельзя. Программа не должна создавать диалоговые окна (как графические, так и текстовые), подгружать другие модули и библиотеки и т. п.

4. Ресурсы и ограничения в процессе решения задач.

Участники могут пользоваться Java API Documentaion, доступной по [ссылке](#). Никакими другими ресурсами пользоваться нельзя. Участники ни при каких обстоятельствах не могут принимать помощь, советы, решения от третьих лиц.

5. Описание тестовой системы

Решения могут быть проверены жюри на другом наборе данных, нежели тот, на котором эти решения отлаживали участники. Доступ к системе проверки решений осуществляется через сеть Интернет – каждый участник сам записывает в систему свое решение. Каждый участник самостоятельно контролирует успешность записи его решения в систему проверки.

После того, как Вы нажмете на кнопку отправить вверху страницы изменится статус. При появлении вверху страницы кнопки "Обновить" нажмите на нее, чтобы увидеть результат проверки.

Задание считается принятым, если решение прошло все тесты в тестирующей системе. В случае ошибки участник получает вердикт тестирования и, если это имеет смысл, номер теста который задание не прошло

Все возможные вердикты тестирования перечислены ниже:

0 ОК ОК ОК

1 CE Compilation Error ошибка компиляции

2 RT Run-Time Error ошибка при работе программы

3 TL Time Limit Exceeded ошибка превышения лимита времени

4 PE Presentation Error ошибка неправильного формата результата

5 WA Wrong Answer неправильный ответ

6 CF Check Failed внутренняя ошибка проверки

7 PT Partial solution Частичное решение

8 AC Accepted Принято на проверку

9 IG Ignored Решение проигнорировано

10 DQ Disqualified Решение дисквалифицировано

11 PD Pending Ожидает проверки. Выставляется в случае отключения автоматической проверки

12 ML Memory Limit Exceeded ошибка превышения лимита памяти

13 SE Security Violation ошибка нарушения ограничений безопасности

14 SV Style Violation ошибка нарушения стиля оформления исходного код

Результаты соревнований отображаются в реальном времени в таблице результатов.

За 1 час до конца турнира таблица результатов перестанет обновляться, то есть участники будут видеть положение на момент начала заморозки.

6. Примеры задач

Задачи будут двух типов: алгоритмические и на знание Java API.

Задача 1 (алгоритмическая):

Стоимость одного подъема на горнолыжном курорте «Белоснежный» 51 рубль, билет на 10 подъемов стоит 451 рубль, билет на 30 подъемов стоит 1211 рублей, билет на 50 подъемов стоит

1521 рубль. Опытный лыжник Семен собирается съехать с горы n раз. Определите, сколько абонементов каждого вида он должен приобрести, чтобы суммарное количество оплаченных подъемов было не меньше n , а общая стоимость приобретенных абонементов — минимальна.

Программа получает на вход одно число n и должна вывести четыре целых числа, равных необходимому количеству билетов на 1, на 10, на 30, на 50 подъемов.

Если для какого-то n существует несколько способов приобретения абонементов одинаковой суммарной стоимости, необходимо вывести ту комбинацию билетов, которая дает большее число поездок.

Задача 2 (JavaAPI)

На стандартный ввод подается путь к jar файлу в формате текущей операционной системы.

Например.

windows: c:/Temp/sample.jar

unix: /tmp/sample.jar

Jar файл содержит внутри себя набор файлов. Некоторые из них могут содержать байт код для Java классов, интерфейсов и перечислений (enum).

Все файлы с байт кодом имеют расширение .class. Не все файлы, имеющие расширение .class являются валидными class файлами, такие файлы нужно игнорировать.

Необходимо посчитать какое количество классов, интерфейсов и перечислений (enum) содержится в данном jar файле.

Ответ должен быть выведен на стандартный вывод в формате:

a b c

где a - количество классов

где b - количество интерфейсов

где c - количество перечислений (enum)

В случае, если jar файл не является корректным jar файлом, на стандартный вывод необходимо вывести -1.