

DRAFT
Version 0.1 [8.03.2004]

ПРОТОКОЛ КЛИЕНТ-СЕРВЕРНОГО ВЗАИМОДЕЙСТВИЯ ПОД КОДОВЫМ НАЗВАНИЕМ
OLYMP

Илья Солдатенко

Март 2004

Центр Новых Информационных Технологий
Тверской государственной университет

(0822) 36-57-43

Оглавление

1. Каналы связи
2. Состояния сервера
3. Теория кодирования откликов сервера
4. Сводная таблица запросов и ответов
5. Описание запросов
 - a. Синтаксис запросов
 - b. Описание запросов
6. Описание ответов
 - a. Синтаксис ответов
 - b. Описание ответов
7. Структуры данных

1. Каналы связи

Общение между клиентом-участником и сервером, тестером и сервером, администратором и сервером происходит через виртуальный канал связи. Канал связи между клиентом и сервером устанавливается в любой момент сетевого соединения по инициативе клиента.

Каналы связи бывают следующих типов:

1. виртуальный канал для общения клиента-участника и сервера – КУ-канал
2. виртуальный канал для общения тестера и сервера – Т-канал
3. виртуальный канал для общения администратора и сервера – А-канал

Основная идея заключается в том, что каждому каналу соответствует свой круг клиентских запросов, которые могут быть приняты и обработаны сервером, только если они были посланы через соответствующий канал. Например, запустить олимпиаду можно только по А-каналу связи, а запросить тесты и решения – по Т-каналу. Даже административный канал связи не обладает всей полнотой запросов, так как это привело бы к излишней загруженности администратора ненужной функциональностью.

Канал связи устанавливается клиентом с помощью запроса серверу LOGIN. Параметр запроса указывает на тип создаваемого канала. Основными условиями открытия виртуального канала связи служат:

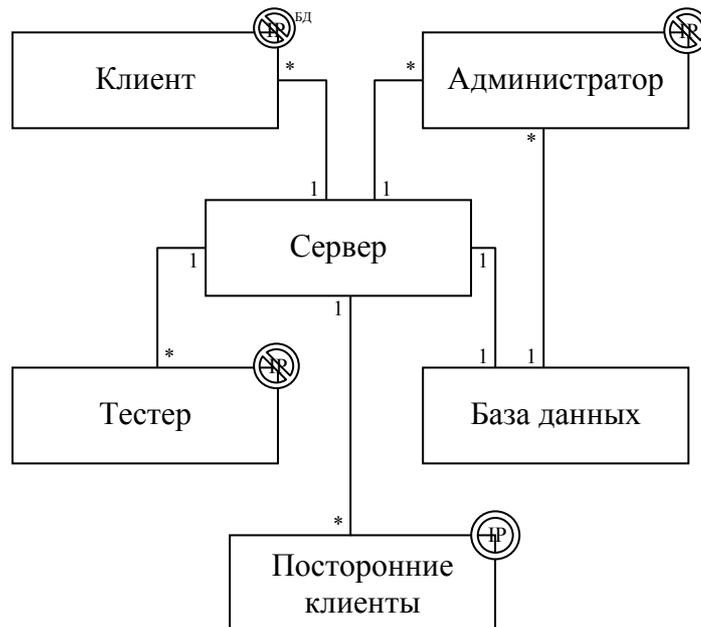
1. для КУ-каналов – клиент должен находиться на машине с IP адресом, зарегистрированным за одной из команд. IP адреса хранятся в базе данных и присваиваются каждой команде индивидуально. Помимо этого, заголовок Code запроса должен содержать кодовое слово, принадлежащее данной команде. В случае, если команда дисквалифицирована, сервер не открывает канал.
2. для Т-каналов – клиент должен находиться на машине с IP адресом, прописанном в конфигурационном файле сервера, как IP адрес тестирующей машины.
3. для А-каналов - клиент должен находиться на машине с IP адресом, прописанном в конфигурационном файле сервера, как IP адрес администраторской машины.

В случае если клиент попытается сделать запрос, запрещенный в данном канале связи, сервер вернет ошибку несоответствия запроса каналу связи.

Канал связи может быть закрыт клиентом либо посредством запроса LOGOUT, либо при закрытии сетевого соединения.

2. Состояния сервера

Общую схему клиент-серверного взаимодействия можно описать следующей диаграммой:



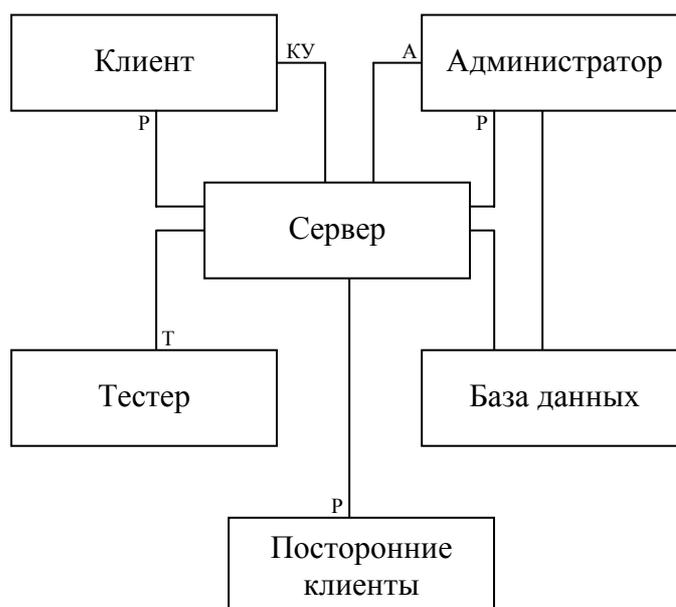
где:

1. – доступ разрешен с любого IP адреса
2. - доступ разрешен только с тех IP адресов, что прописаны в конфигурационном файле сервера
3. - доступ разрешен только с тех IP адресов, что прописаны в базе данных

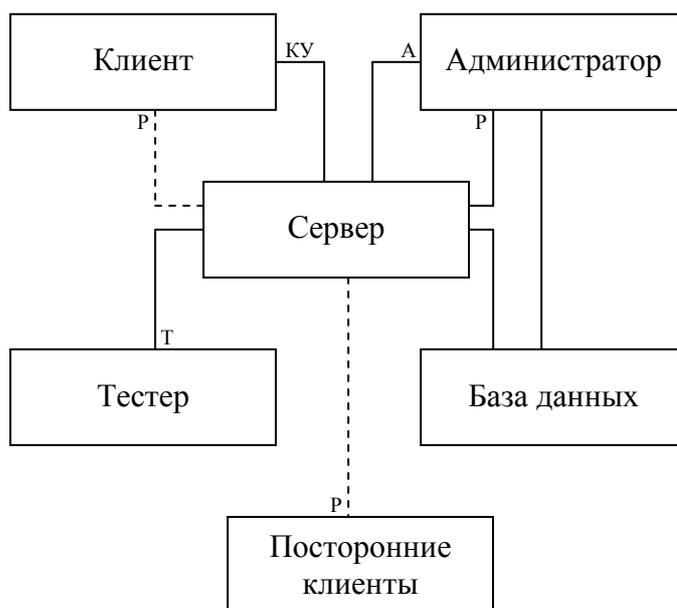
Таким образом, напрямую с базой данных могут общаться только сервер и администратор.

Сервер может быть в одном из следующих состояний:

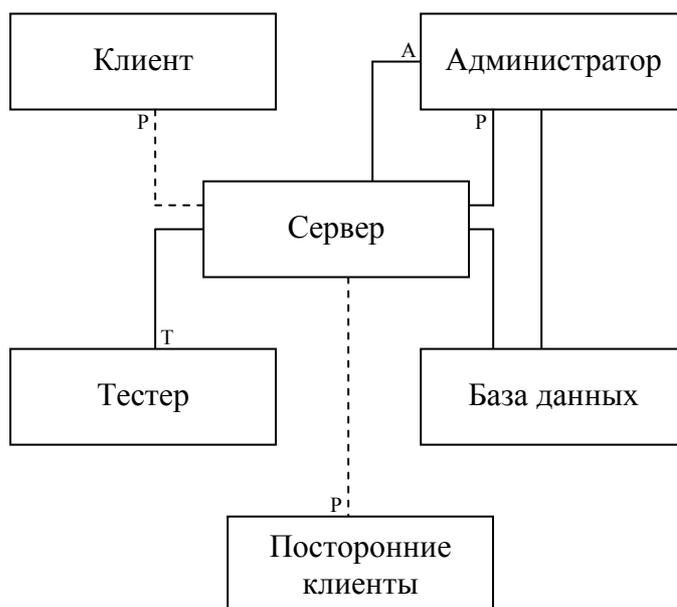
- 1) олимпиада идет (КУ, А, Т – соответствующие каналы передачи данных, Р – активный рейтинг)



2) олимпиада идет, рейтинг заморожен



3) олимпиада остановлена



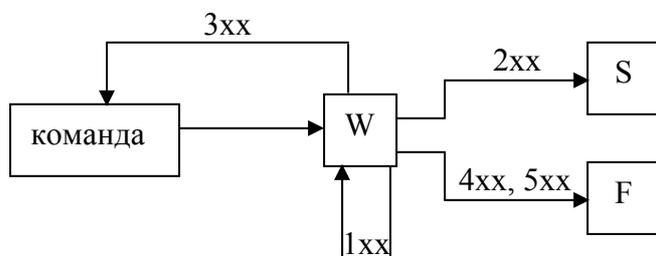
3. Теория кодирования откликов сервера

Код отклика сервера является трехзначным числом, в котором первая цифра идентифицирует тип отклика, а остальные две образуют порядковый номер отклика.

Существует пять значений для первой цифры кода отклика:

- 1yz – позитивный предварительный отклик:
Сюда относятся отклики двух типов:
 - запрос был принят, но его исполнение временно приостановлено в ожидании каких-либо действий от клиента. Клиент должен послать еще один запрос, указывающий, необходимо ли продолжать выполнение первоначальной команды или прервать его.
 - запрос принят и успешно выполнен, но прежде чем отсылать новый запрос, клиент должен дождаться еще одного отклика от сервера.
- 2yx – позитивный заключительный отклик
Запрос был успешно выполнен. Может быть инициирован новый запрос.
- 3yz – позитивный промежуточный отклик
Запрос был принят, но его исполнение временно приостановлено в ожидании дополнительной информации от клиента. Клиент должен послать следующий запрос с необходимой серверу информацией. Данный тип откликов используется в составных запросах, когда одна команда выполняется посредством некоторой цепочки запросов и ответов.
- 4yz – временно-отрицательный заключительный отклик
Запрос был отвергнут, и запрашиваемая команда, соответственно, не была выполнена. Тем не менее, ошибка, из-за которой запрос не был принят на исполнение, не является фатальной, и клиент может повторить запрос, изменив, быть может, некоторые его параметры или подождать некоторое время. Точное значение “временности” зависит от конкретного запроса.
- 5yz – перманентно-отрицательный заключительный отклик
Запрос был отвергнут, и запрашиваемая команда, соответственно, не была выполнена. Клиент не должен посылать тот же самый запрос, так как его обработка невозможна из-за неустранимой ошибки (такой ошибки, которая не может быть устранена сервером самостоятельно, и сервер не имеет никакой дополнительной информации о том, когда и как она будет устранена).

Взаимоотношения откликов сервера можно изобразить на следующей диаграмме:



Здесь:

- команда – команда, посылаемая клиентом серверу на выполнение
- W – состояние ожидание отклика сервера
- S – успешное выполнение команды
- F – неуспешное выполнение команды

4. Сводная таблица запросов и ответов

Запросы	Ответы
LOGIN	100 Wait For Beginning
LOGOUT	101 Program Accepted For Testing
GET-TASKS	102 Free Tester Registered
TASK	200 Logged In
GTT	201 Logged Out
READY	202 Result Of Testing
DONE	203 Tests And Tasks
START	204 Result Accepted
STATUS-CHANGE	205 OK
DSQ	206 Full Rating
REACTIVATE	207 Part Of Rating
INIT	208 Rating Not Changed
RATING-UPDATE	209 Olympiad Started
RATING	210 Already Solved
RATING-PART	211 Tasks
	220 <имя сервера> at <имя хоста>
	300 Reactivate Tests And Tasks
	301 Program For Testing
	400 Forbidden
	401 Method Not Allowed
	402 Team Disqualified
	403 Length Required
	404 Bad Request
	405 Code-Team Disparity
	500 Internal Server Error
	501 Method Not Implemented
	502 OLYMP Version Not Supported

5. Запросы

5.1. Синтаксис запроса

Запрос имеет следующий синтаксис в БНФ-нотации:

запрос := команда [заголовки] [тело]

команда := <команда> [<SP> <параметр>] <SP> <протокол>/<версия>

<команда> := <КУ-запрос> | <Т-запрос> | <А-запрос> | <Ост. запросы>

<КУ-запрос> := TASK

<Т-запрос> := GTT | READY | DONE

<А-запрос> := START | STATUS-CHANGE | DSQ | REACTIVATE | INIT | RATING-UPDATE

<Ост. запросы> := LOGIN | LOGOUT | GET-TASKS | RATING | RATING-PART

<параметр> := {для каждой команды могут быть свои}

<протокол> := {название протокола}

<версия> := {версия протокола}

заголовки := <CRLF> заголовок | заголовки

заголовок := <Имя>: <значение>

тело := <CRLF><CRLF> {текст, предусмотренный запросом}

Регистр команд, параметров, заголовков, а также названия протокола не важен. Тело запроса отделяется от заголовков двумя переносами строк. Команда представляет собой последовательность из латинских букв и дефиса. Порядок заголовков не имеет значения, имя заголовка отделяется от его значения двоеточием.

5.2. Описание команд

Во всех диаграммах состояния, приводимых для запросов, используются следующие обозначения:

- В – начало команды
- W – состояние ожидания клиентом ответа от сервера
- S – успешное завершение команды
- E – неуспешное завершение команды
- C – состояние ожидания сервером запроса от клиента. Встречается в изображении диаграмм составных команд

Свободные запросы и запрос от команды-участницы (КУ-запрос)

1) LOGIN

Назначение: открытие защищенного канала передачи данных.

Параметры:

- team** – открытие КУ-канала, канала между сервером и клиентом-участником олимпиады
- tester** – открытие Т-канала, канала между сервером и тестером
- admin** – открытие А-канала, канала между сервером и администратором

Заголовки:

- Code – кодовое слово команды. Заголовок посылается клиентом только в том случае, если задан параметр team.

Описание:

Общение с сервером может происходить двумя способами – через защищенный канал передачи данных и без него. Каналов связи бывает три – КУ-канал, Т-канал и А-канал (подробнее см. в разделе “Каналы связи”). Для установления канала связи служит запрос LOGIN, причем его параметр указывает на то, какой именно канал клиент хочет установить.

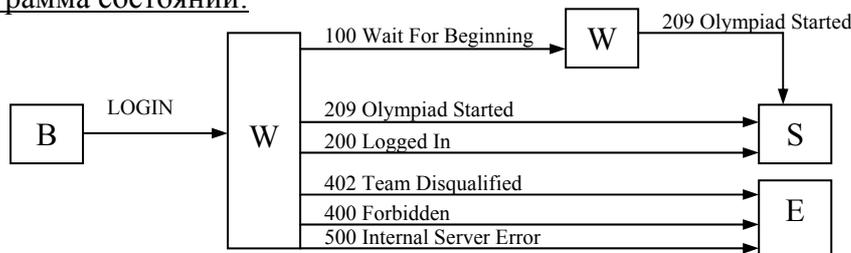
В случае если запрос производится с машины, IP адрес которой не прописан ни в базе данных, ни в конфигурационном файле, то сервер возвращает отклик 400 (*Forbidden*).

Если производится открытие КУ-канала с разрешенного IP адреса, но кодовое слово команды не соответствует кодовому слову той команды, что в действительности должна сидеть за данной машиной, то сервер возвращает ответ 405 (*Code-Team Disparity*).

Если открытие канала происходит успешно, то ответ сервера зависит от типа открываемого канала:

- если открыт КУ-канал, то:
 - если олимпиада еще не началась, сервер посылает клиенту ответ 100 (*Wait For Beginning*). Затем, когда олимпиада начинается, сервер посылает клиенту сообщение 209 (*Olympiad Started*)
 - если олимпиада началась, сервер посылает 209 (*Olympiad Started*)
- если открыт Т-канал или А-канал, то сервер посылает 200 (*Logged In*).

Диаграмма состояний:



Следует заметить, что КУ-канал может быть успешно открыт только в том случае, если олимпиада уже идет, поэтому ответ сервера, завершающий команду открытия КУ-канала (209) отличается от ответа, завершающего открытие А- и Т-каналов (200). Если же клиент попытается отослать задачу при помощи запроса TASK еще до начала олимпиады, то сервер снова возвратит ответ 100 (*Wait For Beginning*).

Если делается попытка установить КУ-канал, а команда-участница является дисквалифицированной, то сервер возвращает соответствующий код ошибки (402).

2) LOGOUT

Назначение: закрытие защищенного канала связи

Параметры: нет

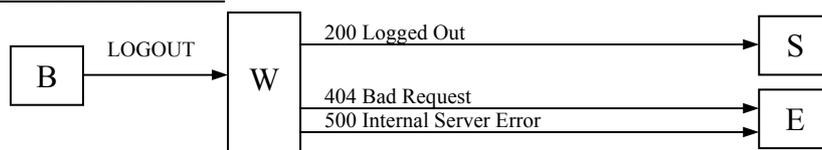
Заголовки: нет

Описание:

Клиент может закрыть канал с сервером по своему желанию, пошлав ему запрос LOGOUT. Данный запрос не имеет ни параметров, ни заголовков. В случае успешного закрытия канала, сервер посылает ответ 201 (*Logged Out*). В случае если клиент посылает данную команду при закрытом канале, сервер возвращает сообщение об ошибочном запросе 404 (*Bad Request*).

Тем не менее, использование команды LOGOUT для закрытия канала передачи данных не является обязательным. Запрос нужен для того, чтобы клиент имел возможность закрыть канал, не закрывая при этом сетевое соединение.

Диаграмма состояний:



3) GET-TASKS

Назначение: запрашивает у сервера номера и названия задач

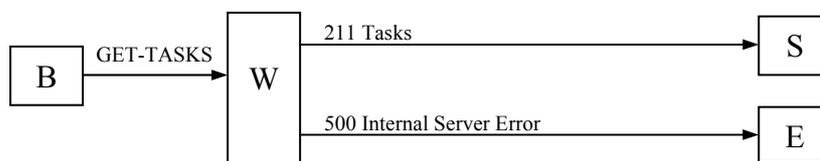
Параметры: нет

Заголовки: нет

Описание:

При помощи данной команды клиент может запросить у сервера номера и названия задач текущей олимпиады. Запрос разрешен в любом канале. В случае успешного выполнения сервер возвращает ответ 211 (*Tasks*), содержащий запрашиваемую информацию.

Диаграмма состояний:



4) TASK

Назначение: отправляет задачу серверу на проверку

Параметры:

- **force** – если программа, отправляемая клиентом, соответствует задаче, которая уже была успешно решена командой ранее, то сервер возвращает ответ 210 (*Already Solved*) и не ставит ее в очередь на проверку, чтобы не нагружать тестеров. Если же клиент все же хочет проверить корректность данной программы, то он может указать в запросе параметр *force*, который заставит сервер протестировать данную задачу и вернуть результат. Но, несмотря на это, тестирование данной задачи никак не отразится на рейтинге команды (даже в том случае, если результат тестирования окажется отрицательным).

Заголовки:

- Task-Id: идентификатор задачи, являющийся уникальным для всех задач, отправляемых клиентом. Идентификатор необходим клиенту для определения того, к какой задаче относится результат, присланный сервером. После проверки задачи, сервер присылает отклик 202 (*Result Of Testing*), одним из заголовков которого является идентичное поле Task-Id. Данное поле будет иметь такое же значение, если результат является решением соответствующей задачи
- Task: порядковый номер задачи
- Content-Length: длина текста программы в байтах

Описание:

При помощи данного запроса клиент может отправить задачу серверу на проверку. Запрос является защищенным и может быть послан только по КУ-каналу.

В теле запроса идет сам текст программы. Необходимо, чтобы реальный размер программы совпадал с тем, что указан в заголовке Content-Length, так как никаких явных

меток, указывающих на конец программы, нет. Если клиент не указал заголовок Content-Length, сервер должен ответить сообщением об ошибке 403 (*Length Required*).

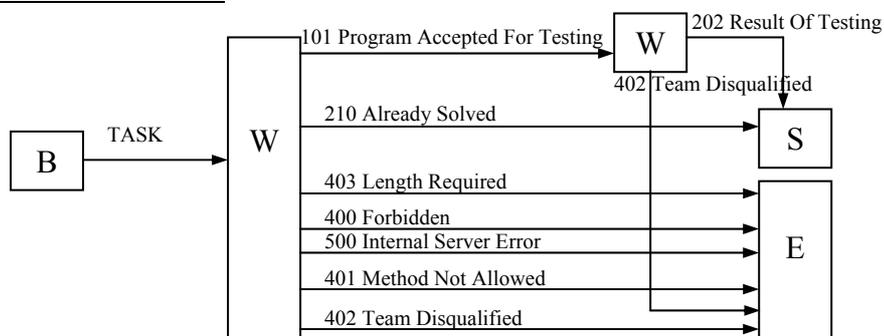
Если задача уже была ранее успешно решена командой, то сервер сразу возвращает ответ 210 (*Already Solved*). Предполагается, что сервер не будет отсылать на тестирование данную задачу, так как это приведет к лишней нагрузке тестера, тем более что данная задача никак не может повлиять на текущий рейтинг. Тем не менее данная задача регистрируется сервером в базе данных

В случае если сервер успешно принял задачу и поставил ее в очередь на проверку, он должен вернуть клиенту отклик 101 (*Program Accepted For Testing*). После того как программа протестирована, сервер возвращает отклик 202 (*Result Of Testing*), содержащий результаты тестирования данной задачи. Спецификация не накладывает никаких ограничений на то, может ли клиент сразу после отсылки первой задачи, отсылать следующую. Так как запрос 101 принадлежит к классу 1xx запросов (См. теорию кодирования откликов сервера), то подразумевается, что выполнение команды еще не завершено и клиент находится в состоянии ожидания ответа от сервера. Все зависит от того, может ли клиент инициировать выполнение параллельной команды и корректно отслеживать выполнение всех выполняющихся в данный момент команд. Предполагается, что сервер умеет обрабатывать параллельные команды, к тому же протокол позволяет это делать (заголовок Task-Id команды TASK предоставляет возможность идентифицировать различные команды).

Запрос является защищенным и может отсылаться только по КУ-каналу. Если это не так, то сервер возвращает сообщение об ошибке 401 (*Method Not Allowed*).

Если между принятием от клиента решения и его проверкой тестером команда дисквалифицируется, то сервер вместо результатов тестирования посылает код ошибки 402 (*Team Disqualified*) и закрывает соединение с клиентом.

Диаграмма состояний:



Запросы от тестера (Т-запросы)

5) GTT

Назначение: запрашивает у сервера номера и временные лимиты задач, а также все тесты и решения к ним

Параметры: нет

Заголовки:

- Tasks: необязательный заголовок, содержащий номера задач, тесты и решения к которым необходимо выслать. Разделены точкой с запятой. Если заголовок не указан, сервер высылает все задачи с тестами и решениями.

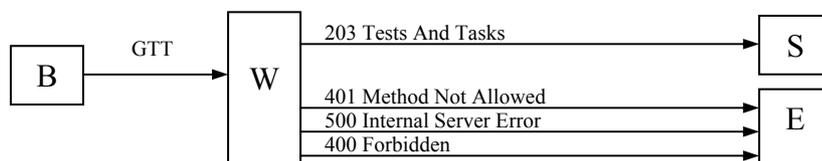
Описание:

При помощи данного запроса тестеры в момент начала своей работы получают у сервера все задачи с лимитом времени, а также все тесты и решения к ним. Тестер может запросить

отдельные задачи, используя заголовок `Tasks` (это может понадобиться в момент реактивации задач – подробнее см. команду `REACTIVATE`).

Данный запрос является защищенным и возможен только в Т-канале. Использование его в других каналах приводит к возврату сервером сообщения об ошибке 401 (*Method Not Allowed*). В случае успешного выполнения сервер возвращает ответ 203 (*Tests And Tasks*) с запрашиваемой информацией.

Диаграмма состояний:



6) `READY`

Назначение: сообщает серверу о готовности сервера принять очередную задачу на проверку

Параметры: нет

Заголовки: нет

Описание:

Открытие Т-канала еще не позволяет серверу отправлять тестеру задачи на проверку. Во-первых, из-за того, что возможно тестеру еще требуется запросить у сервера задачи с тестами и, следовательно, он еще не готов что-либо проверять. Во-вторых, сервер сам не может ничего запрашивать, так как единственный способ его общения с клиентами – это выполнение команд по схеме запрос-ответ.

Именно для решения этих двух проблем и служит запрос `READY`, отсылаемый тестером тогда, когда он готов принимать задачи – к тому же теперь сервер имеет возможность прислать тестеру задачу на проверку вместе с ответом на данный запрос. Запрос `READY` является первым запросом в составной команде `READY-DONE`.

Общая схема выглядит следующим образом. Когда тестер готов принять задачу на проверку, он посылает серверу запрос `READY`. Если с сервером все в порядке, то он сразу же посылает ответ 102 (*Free Tester Registered*) и помещает данного тестера в пул свободных тестеров. Когда сервер решает отправить задачу на проверку, он делает это с помощью ответа 301 (*Program For Testing*) и вдобавок к этому удаляет соответствующего тестера из пула свободных тестеров, таким образом, предотвращая отсылку одновременно двух задач на одну тестирующую машину. Тестер, получив ответ 301, проверяет задачу и возвращает результат проверки серверу с помощью запроса `DONE`. Сервер отвечает на этот запрос сообщением 204 (*Result Accepted*), и на этом общение сервера и тестера заканчивается, так как составная команда `READY-DONE` выполнена полностью. Теперь, для того чтобы получить новую задачу на проверку, тестер должен снова отправить серверу запрос `READY`.

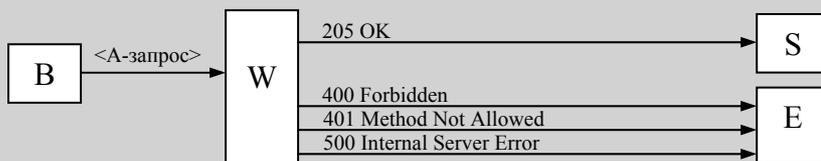
Вместо ответов 102, 301 и 204 сервер может послать ответ 300 (*Reactivate Tests And Tasks*). В этом случае тестер должен перед тем, как послать запрос `READY`, заново запросить задачи и тесты у сервера посредством запроса `GTT` (см. запрос `REACTIVATE`).

Запрос `READY` является защищенным и допускается в Т-канале. Использование его в другом канале приводит к ошибке 401 (*Method Not Allowed*), а использование его вне какого-либо канала – к ошибке 400 (*Forbidden*).

Администраторские запросы (А-запросы)

Все администраторские запросы в случае успешного их выполнения получают от сервера один и тот же ответ – 205 (OK).

Все А-запросы также имеют одинаковую диаграмму состояний:



8) START

Назначение: запускает олимпиаду

Параметры: нет

Заголовки: нет

Описание:

Администраторский запрос, запускающий олимпиаду. Получив данный запрос, сервер переходит в активное состояние (см. Состояния сервера), посылает всем уже подключенным КУ-клиентам ответ 209 (*Olympiad Started*) и начинает прием запросов TASK и их обработку. Помимо этого сервер также берет на себя обязанность изменить статус олимпиады в базе данных.

В случае успеха сервер возвращает ответ 205 (OK).

Запрос START является защищенным и допускается в А-канале. Использование его в другом канале приводит к ошибке 401 (*Method Not Allowed*), а использование его вне какого-либо канала – к ошибке 400 (*Forbidden*).

9) STATUS-CHANGE

Назначение: меняет статус олимпиады

Параметры:

- **freeze** – заморозить рейтинг
- **melt** – разморозить рейтинг
- **stop** – остановить олимпиаду

Заголовки: нет

Описание:

Позволяет вручную изменить статус олимпиады. В нормальной ситуации единственным ручным вмешательством в проведение олимпиады является ее запуск посредством запроса START. Сервер имеет параметр конфигурации – t (считываемый из конфигурационного файла в момент запуска сервера), определяющий интервал времени до конца олимпиады, когда замораживается рейтинг.

В момент, когда <время окончания олимпиады> - <текущее время олимпиады> $\leq t$, рейтинг замораживается автоматически. Когда время олимпиады выходит, она также останавливается автоматически.

Если же требуется ручное вмешательство в текущий ход олимпиады, то используется команда STATUS-CHANGE с одним из указанных параметров, но при этом происходят следующие необратимые вещи:

- в случае заморозки рейтинга (*freeze*), параметру *t* присваивается текущая продолжительность олимпиады, т.е. время <время окончания олимпиады> - <текущее время олимпиады>. Таким образом, рейтинг становится замороженным. (Пржнее значение параметра *t* при этом сохраняется сервером).
- в случае разморозки рейтинга (*melt*), параметру *t* присваивается значение 0. Таким образом, рейтинг размораживается до конца проведения олимпиады. Таким же способом можно разморозить рейтинг тогда, когда он уже был заморожен сервером.

Восстановить исходное значение параметра *t* можно при помощи запроса *INIT*. Как и в случае запроса *START* сервер также должен самостоятельно внести изменения в базу данных, касающиеся состояния сервера (в случае остановки сервера – записать новое значение статуса олимпиады).

В случае успеха сервер возвращает ответ 205 (*OK*).

Запрос *START* является защищенным и допускается в А-канале. Использование его в другом канале приводит к ошибке 401 (*Method Not Allowed*), а использование его вне какого-либо канала – к ошибке 400 (*Forbidden*).

10) DSQ

Назначение: дисквалифицирует команду

Параметры: нет

Заголовки:

- *Team*: код дисквалифицируемой команды

Описание:

Позволяет дисквалифицировать команду в момент проведения олимпиады. Помечает команду как дисквалифицированную, так что на любой запрос от данной команды, сервер будет возвращать 402 (*Team Disqualified*). Сервер также должен сам внести изменения в базу данных, касающиеся дисквалифицированной команды.

В случае успеха сервер возвращает ответ 205 (*OK*).

Запрос *START* является защищенным и допускается в А-канале. Использование его в другом канале приводит к ошибке 401 (*Method Not Allowed*), а использование его вне какого-либо канала – к ошибке 400 (*Forbidden*).

11) REACTIVATE

Назначение: заставляет сервер послать всем активным тестерам сообщение о реактивации

Параметры: нет

Заголовки:

- *Tasks*: номера задач, которые необходимо реактивировать. Разделяются точкой с запятой. Если заголовок не указан, то реактивируются все задачи.

Описание:

В случае если вносятся какие-либо изменения в задачи, тесты или ответы к ним, то необходимо, чтобы тестеры узнали об этих изменениях. Для этого администратор посылает серверу данный запрос. После этого сервер посылает всем активным тестерам сообщение о том, что им следует заново загрузить указанные (или все) задачи. Соответственно пул свободных тестеров освобождается.

В случае успеха сервер возвращает ответ 205 (*OK*).

Запрос REACTIVATE является защищенным и допускается в А-канале. Использование его в другом канале приводит к ошибке 401 (*Method Not Allowed*), а использование его вне какого-либо канала – к ошибке 400 (*Forbidden*).

12) INIT

Назначение: заставляет сервер перезагрузить все свои динамические настройки из базы данных

Параметры: нет

Заголовки: нет

Описание:

Все настройки сервера можно разделить на две группы: динамические – те, которые загружаются из базы данных, и статические – которые загружаются из конфигурационного файла в момент запуска сервера.

К числу статических настроек относятся прежде всего IP адреса машин-администраторов и машин-тестеров.

К числу наиболее важных динамических настроек относятся следующие вещи:

- код олимпиады. (Обычно последняя по дате проведения олимпиада является активной. Используется во всех запросах в базе данных для идентификации текущей олимпиады.)
- статус (остановлена, идет, выгружена)
- продолжительность
- пробный тур
- список всех команд – их код, статус, IP адрес, кодовое слово

В момент, когда загружается сервер, он считывает динамические настройки из базы данных. Далее могут возникнуть моменты, когда необходимо их перезагрузить. Для этого можно либо перезагрузить сам сервер, если надо еще обновить и статические параметры, либо использовать запрос INIT. Например, для того чтобы создать новую олимпиаду, нужно занести в базу данных всю информацию о ней, а затем послать серверу запрос INIT, после чего сервер начнет работать с вновь созданной олимпиадой.

Подразумевается, что запрос не приводит к уничтожению и созданию всех внутренних объектов сервера, так как данный запрос должен без какого-либо ущерба работать и тогда, когда олимпиада уже идет (например, если машина команды-участницы выйдет из строя и необходимо будет выдать ей новую машину, а, следовательно, и новый IP адрес).

Можно заметить, что запросы START, STATUS-CHANGE, DSQ являются своего рода “точечными” запросами, изменяющими лишь отдельные параметры динамической конфигурации сервера. Все, что делают данные запросы, можно сделать также с помощью одного запроса INIT – для этого необходимо вручную изменить соответствующие записи в базе данных и послать серверу запрос INIT.

В случае успеха сервер возвращает ответ 205 (*OK*).

Запрос INIT является защищенным и допускается в А-канале. Использование его в другом канале приводит к ошибке 401 (*Method Not Allowed*), а использование его вне какого-либо канала – к ошибке 400 (*Forbidden*).

13) RATING-UPDATE

Назначение: заставляет сервер актуализировать рейтинг

Параметры: нет

Заголовки: нет

Описание:

Для того чтобы получить рейтинг, существуют два запроса: RATING и RATING-PART. Если рейтинг не заморожен, то данные для формирования ответов на эти запросы берутся непосредственно из соответствующих таблиц базы данных. В момент, когда рейтинг замораживается, сервер делает “снимок” (т.е. текст рейтинга в том же самом формате, в котором сервер отправляет его клиенту) текущего рейтинга и размещает его в памяти. При получении всех дальнейших запросов на выдачу рейтинга, сервер не обращается к базе данных, а отправляет этот самый “снимок”. Исключение составляют лишь рейтинговые запросы, отправленные по А-каналу – в этом случае данные все также берутся из базы данных.

Рейтинг является активным только в том случае, если олимпиада идет и время заморозки еще не настало (исключение составляет А-канал – в нем рейтинг всегда активен). Следовательно, когда олимпиада остановлена (в том числе непосредственно после запуска сервера), рейтинг заморожен и существует некоторый “снимок” рейтинга.

Запрос RATING-UPDATE заставляет сервер построить новый “снимок”, исходя из данных, находящихся в базе данных сейчас. Это полезно после окончания олимпиады и оглашения результатов – когда результаты официально оглашены желательно, чтобы рейтинг, возвращаемый сервером, их показывал.

В случае успеха сервер возвращает ответ 205 (OK).

Запрос INIT является защищенным и допускается в А-канале. Использование его в другом канале приводит к ошибке 401 (*Method Not Allowed*), а использование его вне какого-либо канала – к ошибке 400 (*Forbidden*).

Рейтинговые запросы (доступны везде)

14) RATING

Назначение: запрашивает всю таблицу рейтинга

Параметры:

- **with-last-id:** если указать, то сервер помимо таблицы всего рейтинга вернет и идентификатор последней записи в таблице Лог олимпиады

Заголовки: нет

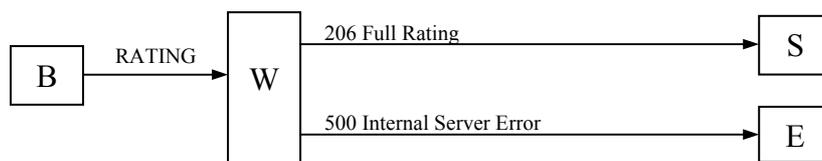
Описание:

Запрашивает всю таблицу рейтинга. При помощи данного запроса реализуется статически обновляемый рейтинг. Клиент не должен рассчитывать на то, что сервер вернет таблицу с уже отсортированными записями.

В случае успеха сервер возвращает ответ 206 (*Full Rating*) с запрашиваемой информацией.

Запрос RATING не является защищенным и допускается везде.

Диаграмма состояний:



15) RATING-PART

Назначение: запрашивает последние записи из таблицы Лог олимпиады

Параметры: нет

Заголовки:

- From: запрашивает все записи из таблицы Лог решений, идентификатор которых больше указываемого в данном заголовке

Описание:

Клиент может реализовывать вывод рейтинга двумя способами: статическим и динамическим. При статическом способе клиент предоставляет пользователю самому решать, когда необходимо загрузить рейтинг. При динамическом – клиент загружает рейтинг автоматически через определенный интервал времени, создавая впечатление вывода рейтинга в режиме реального времени.

Для динамического вывода, клиент использует следующий механизм. Сначала он запрашивает всю таблицу рейтинга, используя запрос RATING с параметром with-last-id. После этого он через фиксированные промежутки времени посылает запрос RATING-PART с имеющимся идентификатором. Запрос RATING-PART помимо данных возвращает также новое значение идентификатора, которое клиент пошлет в следующий раз.

На запрос RATING-PART сервера возвращает не таблицу с рейтингом, а последние записи из таблицы Лог решений, по которым клиент сам восстанавливает картину текущего рейтинга. Если значение, указанное в заголовке From, равно идентификатору последней записи в базе данных (то есть с момента последнего запроса никаких новых данных в базу не было добавлено), то сервер возвращает ответ 208 (*Rating Not Changed*).

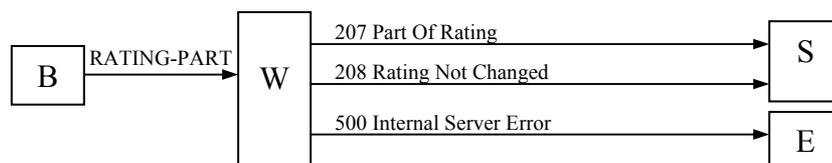
Когда рейтинг заморожен, при получении запроса RATING-PART, сервер ведет себя следующим образом:

- если идентификатор в заголовке From меньше или равен идентификатору (n) последней записи, положенной сервером в таблицу перед заморозкой рейтинга, то сервер возвращает записи, начиная с From + 1 и заканчивая n
- в противном случае, сервер возвращает ответ 208 (*Rating Not Changed*).

В случае успеха сервер возвращает ответ 207 (*Part Of Rating*) с запрашиваемой информацией.

Запрос RATING-PART не является защищенным и допускается везде.

Диаграмма состояний:



6. Ответы

6.1. Синтаксис ответа

Ответ имеет следующий синтаксис в БНФ-нотации:

```
ответ := <ответ> [заголовки] [тело]
ответ := <протокол>/<версия> <SP> <код> <SP> <текст ответа>
<код> := 1xx | 2xx | 3xx | 4xx | 5xx
1xx := 100 | 101 | 102
2xx := 200 | 201 | 202 | 203 | 204 | 205 | 206 | 207 | 208 | 209 | 210 | 211 | 220
3xx := 300 | 301
4xx := 400 | 401 | 402 | 403 | 404 | 405
5xx := 500 | 501 | 502
<текст ответа> := {для каждого кода свой}
<протокол> := {название протокола}
<версия> := {версия протокола}
заголовки := <CRLF> заголовок | заголовки
заголовок := <Имя>: <значение>
тело := <CRLF><CRLF> {текст, предусмотренный ответом}
```

Регистр текста ответа, параметров, заголовков, а также названия протокола не важен. Тело ответа отделяется от заголовков двумя переносами строк. Порядок заголовков не имеет значения, имя заголовка отделяется от его значения двоеточием.

6.2. Описание ответов

1) *100 Wait For Beginning*

Назначение: посылается сервером клиенту при установлении КУ-канала, когда олимпиада еще не началась

2) *101 Program Accepted For Testing*

Назначение: возвращается клиенту на запрос TASK. Указывает на то, что задача успешно поставлена в очередь на проверку

3) *102 Free Tester Registered*

Назначение: возвращается тестеру на запрос READY. Указывает на то, что тестер успешно перемещен в пул свободных тестеров

4) *200 Logged In*

Назначение: сообщает об успешном открытии виртуального канала передачи данных

5) *201 Logged Out*

Назначение: сообщает об успешном закрытии виртуального канала передачи данных

6) *202 Result Of Testing*

Назначение: возвращает клиенту результат тестирования программы

Заголовки:

- Task-Id: идентификатор задачи, являющийся уникальным для всех задач, отсылаемых клиентом. Идентификатор необходим клиенту для определения того, к какой задаче относится результат, присланный сервером
- Result: код результата
 - -1 – задача не проверялась
 - 0 – задача прошла все тесты успешно
 - 1 – ошибка компиляции (compilation error)
 - 2 – превышение предела времени (time limit exceeded) *
 - 3 – нарушение правил соревнований (security violation) *
 - 4 – ошибка исполнения (runtime error) *
 - 5 – нарушение формата вывода (presentation error) *
 - 6 – неверный ответ (wrong answer) *
- Message: опциональный заголовок для вывода текстового сообщения об ошибке.
- Test-Number: для всех результатов, помеченных выше звездочкой, тестер также возвращает порядковый номер теста, на котором возникла ошибка

7) 203 Tests And Tasks

Назначение: возвращает тестеру номера задач, лимиты по времени для каждой задачи, тесты и ответы к ним

Заголовки:

- Tasks-Number: количество задач
- Time-Limits: ограничения по времени ко всем задачам в секундах; идут в порядке возрастания порядковых номеров соответствующих задач. Разделяются точкой с запятой
- Total-Tests: общее количество тестов по всем задачам
- Boundaries: через точку с запятой определяются два параметра
 - 1) test-boundary="<разделитель1>": разделитель1 – строка, ограничивающая текст теста снизу
 - 2) output-boundary="<разделитель2>": разделитель2 – строка, ограничивающая текст ответа к тесту снизу

Описание:

Тело ответа состоит из подряд идущих блоков тестов и ответов к ним:

```
<блок с тестом1>  
разделитель1  
<блок с ответом к тесту1>  
разделитель2  
...  
<блок с тестомN>  
разделитель1  
<блок с ответом к тестуN>  
разделитель2
```

<Блок с тестом> выглядит следующим образом:

```
Task: номер задачи, для которой определен этот тест  
Test: номер теста  
{сам текст теста}
```

<Блок с ответом к тесту> выглядит следующим образом:

Task: номер задачи, для которой определен тест
Test: номер теста, соответствующего данному ответу
{сам текст ответа}

8) *204 Result Accepted*

Назначение: возвращается тестеру в ответ на запрос DONE. Сообщает о том, что результат проверки программы принят

9) *205 OK*

Назначение: возвращается администратору при успешном выполнении A-команды

Заголовки:

- Message: текстовое сообщение о той команде, что была выполнена

10) *206 Full Rating*

Назначение: возвращает всю таблицу рейтинга

Заголовки:

- Teams-Number: количество команд, участвующих в олимпиаде (представленных в рейтинге)
- Tasks-Number: количество задач в олимпиаде

Описание:

Тело ответа содержит Teams-Number строк следующего формата:

<код команды> \t <название команды> \t <рез>₁ \t ... \t <рез>_{Tasks-Number} \t <КРЗ> \t <В>

где:

- \t – символ табуляции
- <код команды> - код команды, как он есть в базе данных
- <рез>_i – запись вида +[N] или –[N], причем знак плюс означает, что задача решена, знак минус – что задача нерешена, N – количество неудачных попыток решить задачу. Всего этих записей столько же, сколько и задач.
- <КРЗ> - количество решенных командой задач
- <В> - время, потраченное на решение задач

11) *207 Part Of Rating*

Назначение: возвращает последние записи из таблицы Лог решений

Заголовки:

- From: идентификатор последней записи в таблице Лог решений (если рейтинг заморожен, то идентификатор последней незамороженной записи)
- Records: количество записей

Описание:

Тело ответа содержит Records строк в следующем формате:

<код команды> \t <номер задачи> \t <тип ошибки> \t <время решения>

где:

- \t – символ табуляции
- <код команды> - код команды, как он есть в базе данных
- <номер задачи> - порядковый номер задачи
- <тип ошибки> - количество решенных командой задач
 - -1 – задача не проверялась
 - 0 – задача прошла все тесты успешно
 - 1 – ошибка компиляции (compilation error)
 - 2 – превышение предела времени (time limit exceeded)*
 - 3 – нарушение правил соревнований (security violation)*
 - 4 – ошибка исполнения (runtime error)*
 - 5 – нарушение формата вывода (presentation error)*
 - 6 – неверный ответ (wrong answer)*
- <время решения> - время, потраченное на решение задачи

12) 208 Rating Not Changed

Назначение: возвращает сообщение о том, что с момента последнего запроса рейтинг не изменялся

Заголовки:

- From: содержит тот же идентификатор, что и соответствующий рейтинговый запрос

13) 209 Olympiad Started

Назначение: возвращается клиенту, открывшему КУ-канал во время проведения олимпиады. Также возвращается клиенту в момент начала олимпиады, если канал уже был открыт ранее

14) 210 Already Solved

Назначение: возвращает сообщение о том, что задача уже была решена командой ранее

Заголовки:

- Task-Number: номер задачи, о которой идет речь

15) 211 Tasks

Назначение: возвращает названия задач

Заголовки:

- Tasks-Number: количество задач

Описание:

Тело ответа содержит Tasks-Number строк, каждая из которых содержит название соответствующей задачи. Все строки упорядочены по номеру соответствующей задачи.

16) *220 <имя сервера> at <имя хоста>*

Назначение: строка приветствия сервера. Возвращается любому клиенту в момент установления сетевого соединения

17) *300 Reactivate Tests And Tasks*

Назначение: Сообщает тестеру о том, что необходимо заново загрузить все тесты и ответы к ним

Заголовки:

- **Tasks:** опциональный заголовок, указывающий, какие задачи тестеру следует загрузить заново. Номер задач идут через точку с запятой. Если заголовок опущен, то заново загрузить необходимо все задачи

18) *301 Program For Testing*

Назначение: Возвращает тестеру программу для тестирования

Заголовки:

- **Task:** порядковый номер задачи, решение к которой находится в теле ответа
- **Content-Length:** длина тела ответа в байтах

Описание:

В теле ответа идет сама программа. Необходимо, чтобы ее размер соответствовал указанному в заголовке.

19) *400 Forbidden*

Назначение: сообщает о том, что запрос не может быть выполнен из-за отсутствия прав доступа

20) *401 Method Not Allowed*

Назначение: возвращается клиенту при попытке послать защищенный запрос, не соответствующий каналу связи

21) *402 Team Disqualified*

Назначение: посылается клиенту в ответ на запросы LOGIN и TASK, если соответствующая команда дисквалифицирована

22) *403 Length Required*

Назначение: возвращается клиенту в ответ на запрос TASK, если в данном запросе опущен заголовок Content-Length

23) *404 Bad Request*

Назначение: возвращается в ответ на ошибочный запрос. Ошибочным считается запрос, у которого синтаксис команды корректен, а в заголовках или теле запроса содержатся ошибки. Также возвращается в ответ на команду LOGOUT, если она послана вне какого-либо канала передачи данных

24) *405 Code-Team Disparity*

Назначение: возвращается при попытке открыть КУ-канал, если указанное в заголовке Code кодовое слово не соответствует кодовому слову команды, находящемуся в базе данных

25) *500 Internal Server Error*

Назначение: возвращается при внутренней ошибке сервера, как например: нет ни одного зарегистрированного тестера, нет подключения к базе данных и т.д.

Заголовки:

- Message: текстовое описание ошибки

26) *501 Method Not Implemented*

Назначение: возвращается сервером, если формат строки запроса корректен, но сама команда запроса не описывается данной спецификацией

27) *502 OLYMP Version Not Supported*

Назначение: возвращается клиенту, если в запросе указан протокол более поздней версии

7. Структуры данных

Олимпиада

- 1) Код олимпиады
- 2) Название олимпиады
- 3) Дата проведения
- 4) Статус (остановлена, идет, выгружена)
- 5) Продолжительность
- 6) Пробный тур (да, нет)

Команда

- 1) Код олимпиады
- 2) Код команды
- 3) Название команды
- 4) Статус (нормальный, дисквалифицирована)
- 5) IP-адрес
- 6) Кодовое слово команды

Участник

- 1) Код команды
- 2) Код участника
- 3) ФИО
- 4) Роль участника (обычный, магистр, тренер, ...)

Задачи

- 1) Код задачи
- 2) Код олимпиады
- 3) Номер задачи
- 4) Название задачи
- 5) Ограничение по времени
- 6) Текст задачи

Тесты

- 1) Код теста
- 2) Код задачи
- 3) Текст теста
- 4) Текст ответа

Лог решений

- 1) Код решения
- 2) Код олимпиады
- 3) Код команды
- 4) Номер задачи
- 5) Тип ошибки
 - -1 – задача не проверялась
 - 0 – задача прошла все тесты успешно
 - 1 – ошибка компиляции (compilation error)
 - 2 – превышение предела времени (time limit exceeded)*
 - 3 – нарушение правил соревнований (security violation)*

- 4 – ошибка исполнения (runtime error) *
 - 5 – нарушение формата вывода (presentation error) *
 - 6 – неверный ответ (wrong answer) *
- 6) Номер теста, с которым задача не справилась
 - 7) Текст решения
 - 8) Время, потраченное на решение

Рейтинг

- 1) Код олимпиады
- 2) Код команды
- 3) Место
- 4) Количество решенных задач

Рейтинг по задачам

- 1) Код олимпиады
- 2) Код команды
- 3) Номер задачи
- 4) Количество неудачных попыток
- 5) Результат (решена, нерешена)
- 6) Время решения