

## Система организации интернет олимпиады по программированию

*Ленкин Алексей Викторович*

*Приамурский государственный университет имени Шолом-Алейхема*

*Студент*

*Лучанинов Дмитрий Васильевич*

*Приамурский государственный университет имени Шолом-Алейхема*

*старший преподаватель кафедры информационных систем, математики и методик обучения*

### Аннотация

В данной статье рассмотрена технология создания web-ориентированного приложения для организации интернет олимпиады по спортивному программированию. Рассмотрены аналоги программного обеспечения.

**Ключевые слова:** PHP, спортивное программирование, интернет олимпиада

### The Internet programming contest organization system

*Lenkin Aleksei Viktorovich*

*Sholom-Aleichem Priamursky State University*

*student*

*Luchaninov Dmitry Vasilyevich*

*Sholom-Aleichem Priamursky State University*

*Senior lecturer of the Department of Information Systems, Mathematics and training methodic*

### Abstract

In the article web-based application technology of creating competitive programming Internet competition organization is described. Software counterparts are considered.

**Keywords:** PHP, competitive programming, Internet competition

Спортивное программирование – самый популярный вид соревнований среди студентов в IT-сфере по всему миру. Соревнование состоит в том, чтобы создать программу под представленные задачи на одном из предлагаемых языков программирования, которая даст верный ответ на всех заготовленных тестах. Олимпиады бывают личные и командные. В командных олимпиадах обычно участвует 3 человека и им на всё время олимпиады предоставляется 1 компьютер для решения задач [1]. Целью участников является решить больше задач и быстрее чем их конкуренты. Самым престижным студенческим соревнованием является чемпионат мира

по программированию (International Collegiate Programming Contest, ICPC), который проводится Ассоциацией вычислительной техники (Association for Computing Machinery, ACM) с 1970 года [2].

Целью исследования является разработка программного обеспечения для организации интернет олимпиады по спортивному программированию.

Для проведение олимпиад по спортивному программированию требуется установить специальные системы на сервер организаторов, которые будут обрабатывать и проверять решения участников и выводить результаты. Также эти системы необходимы для создания турниров, добавления задач и тестов к ним. Самыми известными системами такого типа являются “Contester”, “CATS” и “ejudge”.

Система Contester была создана И.Н.Клоповым на языке Delphi. После развёртывания на сервере позволяет проводить турниры и индивидуальные решения задач по спортивному программированию. Поддерживает большинство современных языков, таких как: языков: C++, Object Pascal, Java и языках .NET: C#, J# и Visual Basic[3]. Работает на Windows и на Linux [4].

На данный момент в Contester заложены правила ACM-соревнований. Турнирная система содержит HTTP-сервер и тестирующий модуль. Работа с системой, чтение текстов задач и отправка решений происходит через веб-браузер.

Программа работает следующим образом [5]: после того, как участник отправил своё решение, происходит компиляция на выбранном языке, затем система проводит программу по тестам, подставляя соответствующие файлы ввода “input.txt” и сразу же генерирует выходные файлы в “output.txt” и сравнивает их с эталонными в файле “pattern.txt”, возвращая результат тестирования. Также Contester сообщает о неудачно скомпилированных программах и может учитывать ограничения по памяти и времени для каждой отдельной задачи. Система была доступна для скачивания бесплатно до 01.09.2016, но на данный момент официальный сайт не доступен и скорее всего система больше не поддерживается разработчиком. Хотя последняя версия системы вышла 07.11.2010, она всё ещё используется рядом учреждений из-за легкой настройки и работы в ней.

Система CATS [6] была создана на базе Кафедры информатики, математического и компьютерного моделирования Дальневосточного федеральный университета в 2004 году. Главные разработчики системы: Рожков Михаил – автор начальной реализации, версии системы с 0.1 по 1.0 и Кленин Александр – руководитель проекта, автор версий системы с 1.1 по текущую. На базе CATS проводятся чемпионаты ACM по Дальневосточному региону, соревнования школьников, тренировки. В отличии от Contester, существует в одном экземпляре на серверах Дальневосточного федерального университета и не доступна для скачивания или покупки. Проверка решений участников работает по схожему с Contester принципу.

ejudge – это система для проведения различных мероприятий, в которых необходима автоматическая проверка программ. Автор системы

ejudge – Александр Чернов. Она бесплатна и доступна для скачивания на официальном сайте [7]. Принцип проверки заданий схож с таким же у Contester, т. е. необходимо подключать файлы ввода и вывода.

Создание системы было произведено с помощью средств скриптового языка программирования PHP, а также базового языка веб-разметки HTML и системы управления базами данных MySQL.

Система была развернута на виртуальном сервере Denwer. Интерфейс для взаимодействия с системой был значительно упрощён для быстрого доступа к ней (рис.1).

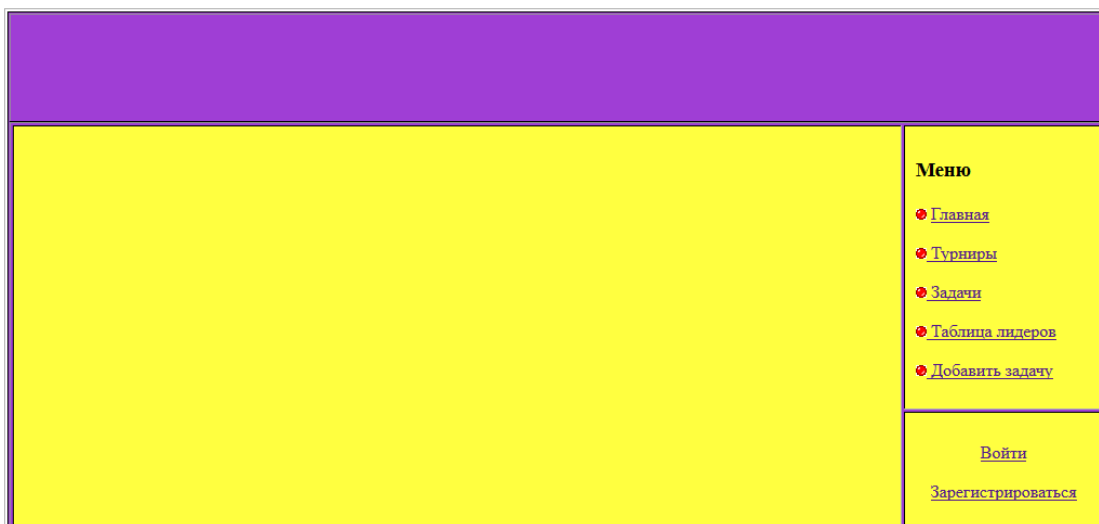


Рис. 1. Интерфейс системы

Для создания регистрации была создана форма обработки данных "register.php", которая получает от пользователя введённые им данные (такие как имя пользователя, логин и пароль) и отправляет их обработчику. На рис. 2 изображена форма регистрации.

The image shows a registration form titled "Регистрация" (Registration). It has a yellow background. The form contains the following fields and elements:

- Имя пользователя: (Username) - text input field with a small icon on the right.
- Логин: (Login) - text input field.
- Пароль: (Password) - text input field with a small icon on the right.
- Повторите пароль: (Repeat password) - text input field with a small icon on the right.
- Зарегистрироваться (Register) - button at the bottom.

Рис. 2. Форма регистрации

После заполнения формы и нажатия кнопки “Зарегистрироваться”, обработчик “register.php” первым делом проверяет заполнены ли все поля, далее проверяет логин на уникальность в базе данных. После прохождения проверки пароль пользователя шифруется в md5 и записывается в базу данных в таблицу users (Рисунок 3.). Код обработчика:

```
<?
$db = mysql_connect ("localhost","disly","1234");
mysql_select_db ("testbase",$db);
if (isset($_POST['submit'])){
$username=$_POST['username'];
$login=$_POST['login'];
$password=$_POST['password'];
$r_password=$_POST['r_password'];
if ($password==$r_password){
$password=md5($password);
$result = mysql_query("SELECT id FROM users WHERE
login='$login'", $db);
$myrow = mysql_fetch_array($result);
if (!empty($myrow['id'])){
exit ("Извините, введённый вами логин уже зарегистрирован.
Введите другой логин.");}
$query=mysql_query("INSERT INTO users VALUES
('','$username','$login','$password','0','-1')");
if ($query=='TRUE'){
mkdir(__DIR__."/users/'".$login.'/',0777,true);
echo "Вы успешно зарегистрированы! Теперь вы можете зайти на
сайт. <a href='index.php'>Главная страница</a>";}
else{echo "Ошибка! Вы не зарегистрированы.";}
else{exit('Пароли не совпадают');}}
?>
```

	id	username	login	password	admin	idturn
<input type="checkbox"/> Изменить  Копировать  Удалить	3	disly	disly	99afb6d7741c5e159d173be768b0a056	0	10
<input type="checkbox"/> Изменить  Копировать  Удалить	8	name	test	99afb6d7741c5e159d173be768b0a056	0	10

↑ Отметить все / Снять выделение С отмеченными: Изменить Удалить Экспорт

Показать : Начальная строка:  Количество строк:  Заголовки каждые  строк

Рис. 3. Таблица “users”

После прохождения регистрации пользователь может авторизоваться в системе, нажав “войти”, и, заполнив форму (рисунок 4), зайти на сайт.

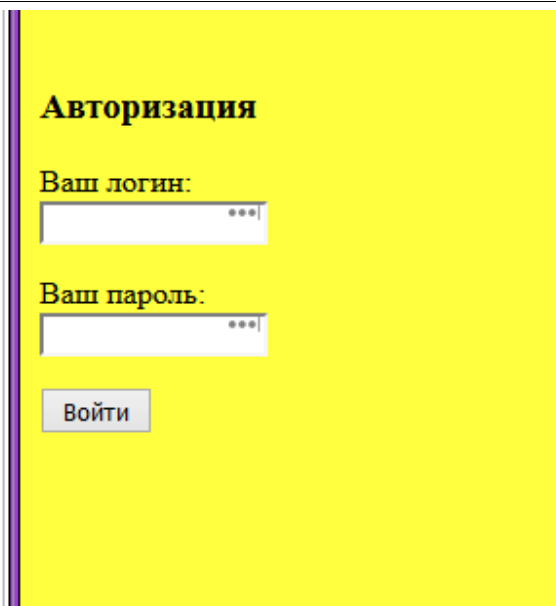


Рис. 4. Форма авторизации

Данные с формы отправляются в обработчик “testreg.php”, где сверяется логин и пароль введённый пользователем с имеющимися в базе данных, и, если авторизация прошла успешно, то каждому посетителю сайта присваивается уникальный идентификатор, называемый идентификатором сессии (session id). Он хранится в cookie на стороне пользователя. Листинг авторизации:

```
<?php session_start();
if (isset($_POST['login'])) { $login = $_POST['login']; if
($login == '') { unset($login);} }
if (isset($_POST['password'])) {
$password=md5($_POST['password']);
if ($password == '') { unset($password);}}
if (empty($login) or empty($password)){
exit ("Вы ввели не всю информацию, вернитесь назад и заполните
все поля!");}
$login = stripslashes($login);
$login = htmlspecialchars($login);
$password = stripslashes($password);
$password = htmlspecialchars($password);
$login = trim($login);
$password = trim($password);
$db = mysql_connect ("localhost","disly","1234");
mysql_select_db ("testbase",$db);
$result = mysql_query("SELECT * FROM users WHERE
login='$login'", $db);
$myrow = mysql_fetch_array($result);
if (empty($myrow['password'])) {
exit ("Извините, введённый вами login или пароль неверный.");}
else {
if ($myrow['password']==$password){
$_SESSION['login']=$myrow['login'];
$_SESSION['id']=$myrow['id'];
```

```

echo "Вы успешно вошли на сайт! <a href='index.php'>Главная
страница</a>"; }
else {
exit ("Извините, введённый вами login или пароль неверный."); } }
?>

```

После авторизации пользователь получает доступ к компонентам системы. Так как по умолчанию турниров не существует, пользователь или администратор должны его создать, открыв вкладку “Турниры” и заполнив табличную форму (рис. 5).

Название турнира:	Время начала турнира:	Время окончания турнира:	Количество задач в турнире:	
Test	2016-12-17 13:00:00	2016-12-17 21:00:00	4	Участствовать
<input type="text"/>	<input type="text" value="YYYY-MM-DD HH:MM:SS"/>	<input type="text" value="YYYY-MM-DD HH:MM:SS"/>	<input type="text"/>	Создать

Рис. 5. Выбор и создание турнира

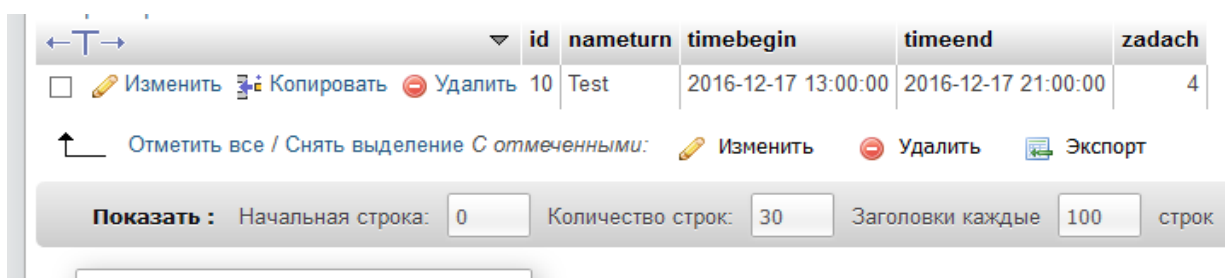
После нажатия кнопки “Создать”, введённая информация отправляется в обработчик “turnirscreate.php”, где после проверки на уникальность турнир записывается в базу данных в таблицу “turnirs”(рис.6), под количество задач создаются папки для хранения тестов и текстов задач, а также создаётся таблица с именем турнира для учёта решений пользователей. Код обработчика:

```

<?
$db = mysql_connect ("localhost","disly","1234");
mysql_select_db ("testbase",$db);
unset($myrow);
if (isset($_POST['submit'])) {
$name = $_POST['nameturn'];
$timebegin = $_POST['timebegin'];
$timeend = $_POST['timeend'];
$zadach = $_POST['zadach'];
$result = mysql_query("SELECT id FROM turnirs WHERE
nameturn='$name'", $db);
$myrow = mysql_fetch_array($result);
if (!empty($myrow['id'])) {
exit ("Извините, введённый вами турнир уже зарегистрирован.
Введите другое название турнира."); }
$query = mysql_query("INSERT INTO turnirs VALUES
('','$name','$timebegin','$timeend','$zadach')");
if ($query == 'TRUE') {
mkdir('zadach/' . $name, 0777, true);

```

```
mysql_query("CREATE TABLE $nameturn (
idusers INT, nameusers TEXT)", $db) Or die(mysql_error());
for ($i=1;$i<=$zadach;$i++){
mkdir('zadach/'.$nameturn.'/'.$i, 0777, true);
$s=strval($i).test;
$s1=strval($i).time;
mysql_query("ALTER TABLE $nameturn ADD $s TEXT")Or
die(mysql_error());
mysql_query("ALTER TABLE $nameturn ADD $s1 TIME")Or
die(mysql_error());}
echo ("Вы создали турнир <a href='turnirs.php'>Вернуться</a>");}
else {
exit ("Ошибка! Турнир не создан. <a
href='turnirs.php'>Вернуться</a>");}}
?>
```



The screenshot shows a web interface for a tournament management system. At the top, there is a search bar and a table with columns: id, nameturn, timebegin, timeend, and zadach. The table contains one row with the following data: id=10, nameturn=Test, timebegin=2016-12-17 13:00:00, timeend=2016-12-17 21:00:00, zadach=4. Below the table, there are several action buttons: Изменить (Change), Копировать (Copy), Удалить (Delete), and Экспорт (Export). At the bottom, there is a pagination control showing 'Показать: Начальная строка: 0, Количество строк: 30, Заголовки каждые 100 строк'.

	id	nameturn	timebegin	timeend	zadach
<input type="checkbox"/>	10	Test	2016-12-17 13:00:00	2016-12-17 21:00:00	4

Рис. 6. Таблица “turnirs”

После создания турнира необходимо заполнить его задачами, для этого необходимо перейти по ссылке “Создать задача” и заполнить форму (рис. 7), выбрав турнир, номер задачи, название задачи, количество тестов, текст задачи и прикрепить тесты в формате “НомерТеста\_in.txt” для входных данных и “НомерТеста\_out.txt” для выходных данных (без символа нижнего пробела).

Название турнира:	Test ▾
Номер задачи:	<input type="text"/>
Название задачи:	<input type="text"/>
Количество тестов:	<input type="text"/>
Текст задачи:	<input type="text"/>
Прикрепить файлы тестов:	<input type="button" value="Обзор..."/> Файлы не выбраны.
<input type="button" value="Создать"/>	

Рис. 7. Форма создания задач

После обработчик “zadachcreatortrue.php” записывает задачу в базу данных в таблицу “zadach” (рис.8), файлы теста в соответствующую папку выбранной задачи и генерирует для неё html-страницу для просмотра задачи и отправки решений. Код обработчика:

```
<?
$turnir=$_POST['turnir'];
$number=$_POST['number'];
$name=$_POST['name'];
$testvalue=$_POST['testvalue'];
$textzadach=$_POST['textzadach'];
$db = mysql_connect ("localhost","disly","1234");
mysql_select_db ("testbase",$db);
$result = mysql_query("SELECT id FROM turnirs WHERE
nameturn='".$turnir'", $db);
$myrow = mysql_fetch_array($result);
$query=mysql_query("INSERT INTO zadach VALUES
('','$number','$name','$testvalue','$myrow[id]')");
$updir='zadach/'.$turnir.'/'.$number.'/';
$cnt = count($_FILES['test']['tmp_name']);
if($cnt > 0){
for($i = 0; $i < $cnt; ++$i){
$tmp_name = $_FILES['test']['tmp_name'][$i];
```








```

$filename = $updir.basename($_FILES['test']['name'][$i]);
move_uploaded_file($tmp_name, $filename);}}
$array=array();
$fp = fopen("up.txt", "r");
while(!feof($fp)) {$array[]=fgets($fp);}
fclose($fp);
$array[]="Задача ".$number;
$fp = fopen("up1.txt", "r");
while(!feof($fp)) {$array[]=fgets($fp);}
fclose($fp);
$array[]=$textzadach;
$array[]="<form action='../.../ctester.php' method='post'>";
$array[]="<input name='zadach' type='hidden'
value=".$number.">";
$array[]="<input name='testnumb' type='hidden'
value=".$testvalue.">";
$fp = fopen("down.txt", "r");
while(!feof($fp)) {$array[]=fgets($fp);}
fclose($fp);
$fp = fopen($updir."text.php", "wb");
$str = '';
foreach($array as $key => $val){$str .= $val;}
file_put_contents($updir."text.php", $str, FILE_APPEND);
echo "Задача создана, <a href='zadachcreator.php'>вернуться<a>";
?>

```

+ Параметры

	id	number	name	test	idturn
<input type="checkbox"/>  Изменить  Копировать  Удалить	30	2	Сложение	3	10

↑ Отметить все / Снять выделение С отмеченными:  Изменить  Удал

Показать : Начальная строка:  Количество строк:  Заголовк

Рис. 8. Таблица “zadach”

Для решения задач пользователю необходимо вначале вступить в турнир на странице “Турниры”, нажав “Участвовать” (рис. 5).

После проделанных действий при нажатии на пункт меню “Задачи” отобразятся все добавленные задачи (рис. 9). Так как при выборе турнира в таблицу пользователя в базе данных был добавлен идентификатор выбранного турнира.

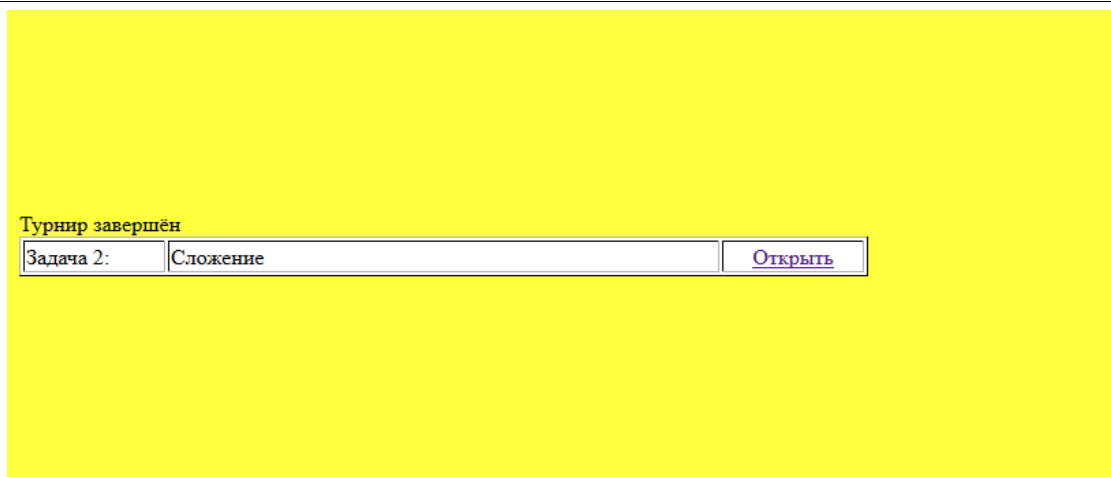


Рис. 9. Список задач турнира

Отображение задач является скриптом и выводит только то, что записано в таблице “zadach” из базы данных. Код отображения:

```
<?
$db = mysql_connect ("localhost","disly","1234");
mysql_select_db ("testbase",$db);
$temp=$_SESSION['login'];
$result = mysql_query("SELECT idturn FROM users WHERE
login='$temp'", $db);
$myrow = mysql_fetch_array($result);
$idid=$myrow['idturn'];
if ($idid==-1)
exit ("Вы не выбрали турнир");
else {
unset($result);
unset($myrow);
date_default_timezone_set("Asia/Yakutsk");
$date1=date('Y-m-d H:i:s');
$result = mysql_query("SELECT * FROM zadach WHERE idturn='$idid'
ORDER BY 'number'", $db);
$res = mysql_query("SELECT * FROM turnirs WHERE
id='$idid'", $db);
$myturn = mysql_fetch_array($res);
$turn=$myturn[1];
$date2=$myturn[2];
$date3=$myturn[3];
$i=0;
if ($date1<$date2) exit ("Турнир ещё не начался <a
href=index.php>На главную</a>");
if ($date3<$date1) echo ("Турнир завершён");
echo "<table bgcolor='FFFFFF' border='1'>";
while ($myrow = mysql_fetch_array($result)){
echo "<tr><td width='100'>Задача ".$myrow[1].":</td><td
width='400'>".$myrow[2]."</td><td align='center' width='100'><a
href=zadach/".$turn."/".$myrow[1]."/text.php>Открыть</a></td></t
r>";}}
?>
```

При нажатии “Открыть” отобразится сгенерированная страница задачи (рис. 10) с формой отправки решений на одном из трёх языков программирования Pascal, C++, C#. Главным преимуществом перед системами такого рода здесь является отправка решений без подключения файлов ввода и вывода.



Рис. 10. Форма отправки решения задачи

При отправке решения в форме (рис.10) код программы отправляется в обработчик “ctester.php”, который является главным модулем системы. Во-первых, он пытается сгенерировать программу на выбранном языке программирование, при ошибке он возвращает “Compilation error” и пользователю засчитывается ошибка в таблицу с именем турнира, а также время ошибки относительно начала турнира. При удачной компиляции скрипт открывает скомпилированную программу в потоке ввода вывода и начинает проверять по тестам, в поток ввода подаются входные файлы теста, а вывод потока сверяется с выходными файлами теста и, если они различны, возвращает “Error in ? test”, где ? – номер теста, записывает ошибку и её время в таблицу. При удачной компиляции и прохождения всех тестов возвращается “Accepted” и записывает удачное решение в таблицу турнира и его время. Код обработчика:

```
<?
session_start();
$zadach=$_POST['zadach'];
$comp=$_POST['comp'];
$prog=$_POST['prog'];
$testnumb=$_POST['testnumb'];
if (strlen($comp)==3){
$fp = fopen(__DIR__."/users/".$_SESSION['login']."/test.cpp",
"w");
fwrite($fp, $prog);
fclose($fp);
$s=__DIR__."/users/".$_SESSION['login']."/test.cpp";
$s1=__DIR__."/users/".$_SESSION['login']."/test";
$s2=__DIR__."/users/".$_SESSION['login'];
```

```
exec("g++ $s -o $s1");}
if (strlen($comp)==2){
$fp = fopen(__DIR__."/users/" .$_SESSION['login'] ."/test.cs",
"w");
fwrite($fp, $prog);
fclose($fp);
$s=__DIR__."/users/" .$_SESSION['login'] ."/test.cs";
$s1=__DIR__."/users/" .$_SESSION['login'] ."/test.exe";
$s2=__DIR__."/users/" .$_SESSION['login'];
exec("csc /out:$s1 /warn:0 /nologo $s");}
if (strlen($comp)==6){
$fp = fopen(__DIR__."/users/" .$_SESSION['login'] ."/test.pas",
"w");
fwrite($fp, $prog);
fclose($fp);
$s=__DIR__."/users/" .$_SESSION['login'] ."/test.pas";
$s1=__DIR__."/users/" .$_SESSION['login'] ."/test";
$s2=__DIR__."/users/" .$_SESSION['login'];
exec("fpc $s");}
$db = mysql_connect ("localhost","disly","1234");
mysql_select_db ("testbase",$db);
$temp=$_SESSION['login'];
$result = mysql_query("SELECT idturn FROM users WHERE
login='$temp'", $db);
$myrow = mysql_fetch_array($result);
$idid=$myrow['idturn'];
$result = mysql_query("SELECT * FROM turnirs WHERE
id='$idid'", $db);
$myrow = mysql_fetch_array($result);
$temp=$myrow['nameturn'];
date_default_timezone_set("Asia/Yakutsk");
$time=strtotime($myrow['timebegin']);
$date1=strtotime(date('Y-m-d H:i:s'));
$hours = floor(($date1-$time)/3600);
$minutes = (($date1-$time)/3600 - $hours)*60;
$seconds = ceil(($minutes - floor($minutes))*60);
$date=$hours.":".floor($minutes).":".$seconds;
$idusers=$_SESSION['id'];
$resulttern = mysql_query("SELECT * FROM $temp WHERE
idusers=$idusers", $db);
$myrowtest=mysql_fetch_array($resulttern);
$vte=$zadach."test";
$vti=$zadach."time";
$valuetest=$myrowtest[$vte];
$valuetime=$myrowtest[$vti];
if
(file_exists(__DIR__."/users/" .$_SESSION['login'] ."/test.exe"))
{for ($test=1;$test<=$testnumb;$test++)
{$f=file(__DIR__."/zadach/" .$_temp ."/" .$_zadach ."/" .$_test ."in.txt"
);
$g=file(__DIR__."/zadach/" .$_temp ."/" .$_zadach ."/" .$_test ."out.txt"
);
$descriptorspec = array(
```

```
0 => array("pipe", "r"), 1 => array("pipe", "w"), 2 =>
array("file", "/tmp/error-output.txt", "a")
$process=proc_open('test', $descriptorspec, $pipes, $s2);
if (is_resource($process)){
foreach($f as $key => $value){
fwrite($pipes[0],$f[$key]);}
fclose($pipes[0]);
$i=0;
while (!feof($pipes[1])){
$answer[$i]=fgets($pipes[1],1024);$i++;}
fclose($pipes[1]);
foreach($answer as $key => $value){
$answer[$key] = htmlspecialchars ( trim($answer[$key]) );
$g[$key] = htmlspecialchars ( trim($g[$key]) );
if ($answer[$key]!=$g[$key]){
unlink(__DIR__."/users/".$_SESSION['login']."/test.exe");
if (empty($valuetest)){
$query=mysql_query("UPDATE $temp SET $vte='-1' WHERE
idusers=$idusers",$db);
$query=mysql_query("UPDATE $temp SET $vti='$date' WHERE
idusers=$idusers",$db)or die(mysql_error());}
else if ($valuetest!='+'){
$valuetest-=1;
$query=mysql_query("UPDATE $temp SET $vte='$valuetest' WHERE
idusers=$idusers",$db) or die(mysql_error());
$query=mysql_query("UPDATE $temp SET $vti='$date' WHERE
idusers=$idusers",$db)or die(mysql_error());}
exit ('Error in '.$test.' test <a href="index.php">Главная
страница</a>');}}
$return_value=proc_close($process);
unset($answer);
unset($f);
unset($g);}}
unlink(__DIR__."/users/".$_SESSION['login']."/test.exe");
if (empty($valuetest)){
$query=mysql_query("UPDATE $temp SET $vte='+' WHERE
idusers=$idusers",$db);
$query=mysql_query("UPDATE $temp SET $vti='$date' WHERE
idusers=$idusers",$db)or die(mysql_error());}
else if ($valuetest!='+'){
$valuetest*=(-1);
$query=mysql_query("UPDATE $temp SET $vte='+$valuetest' WHERE
idusers=$idusers",$db) or die(mysql_error());
$query=mysql_query("UPDATE $temp SET $vti='$date' WHERE
idusers=$idusers",$db)or die(mysql_error());}
echo "Accepted <a href='index.php">Главная страница</a>";}
else {if (empty($valuetest)){
$query=mysql_query("UPDATE $temp SET $vte='-1' WHERE
idusers=$idusers",$db);
$query=mysql_query("UPDATE $temp SET $vti='$date' WHERE
idusers=$idusers",$db)or die(mysql_error());}
else if ($valuetest!='+'){
$valuetest-=1;
```

```

$query=mysql_query("UPDATE $temp SET $vte='$valuetest' WHERE
idusers=$idusers",$db) or die(mysql_error());
$query=mysql_query("UPDATE $temp SET $vti='$date' WHERE
idusers=$idusers",$db)or die(mysql_error());}
exit ('Compilation error <a href="index.php">Главная
страница</a>');}
?>

```

Для просмотра лидеров действующего турнира необходимо открыть ссылку “Таблица лидеров” (рис. 11).

№	Имя	1	2	3	4	V	T
0	test	+ 06:33	+ 06:35			2	788
1	disly	+32 01:42				1	742

Рис. 11. Таблица лидеров

Здесь будет видно: расположение участников в турнирной сетке, их решённые задачи и время их решения относительно начала турнира, общее количество задач и время, затраченное на их решение с учётом штрафного.

Для отображения таблицы лидеров действующего турнира на странице установлен скрипт, который: во-первых, находит в базе данных турнир, который уже начался, но ещё не завершён; во-вторых, получает таблицу этого турнира и обрабатывает её, считая количество решённых задач и штрафное время для решённых задач, которое высчитывается по формуле  $T = \text{время\_затраченное\_на\_решение} + \text{количество\_неудачных\_попыток} * 20$ ; в-третьих, сортирует таблицу турнира по числу решённых задач или, если количество задач такое же как у другого пользователя, то по наименьшему штрафному времени. Код скрипта:

```

<?
$db = mysql_connect ("localhost","disly","1234");
mysql_select_db ("testbase",$db);
$result = mysql_query("SELECT * FROM turnirs",$db);
$b=false;
date_default_timezone_set("Asia/Yakutsk");
$date1=date('Y-m-d H:i:s');
while ($myrow = mysql_fetch_array($result)){
$date2=$myrow[2];
$date3=$myrow[3];

```

```
if ($date1>$date2) {
if ($date3>$date1) {
$b=true;
$zadachvalue=$myrow[4];
$nameturn=$myrow[1];
$idturn=$myrow[0];
break;}}
unset ($myrow);
$result = mysql_query("SELECT * FROM $nameturn", $db);
if ($b) {
$j=0;
$time=array();
$kol=array();
while ($myrow[] = mysql_fetch_array($result)) {
$time[$j]=0;
$kol[$j]=0;
for ($i=2;$i<2*$zadachvalue+2;$i+=2) {
if ($myrow[$j][$i]>0||$myrow[$j][$i]=='+') {
$subs = explode(':', $myrow[$j][$i+1]);
$temp=$subs[0]*60+$subs[1];
$time[$j]+=$myrow[$j][$i]*20+$temp;
$kol[$j]++;}}
$j++;}
for ($k=0;$k<$j;$k++) {
for ($l=0;$l<$j;$l++) {
if ($kol[$l]<$kol[$l+1]) {
$tempkol=$kol[$l];
$kol[$l]=$kol[$l+1];
$kol[$l+1]=$tempkol;
$temptime=$time[$l];
$time[$l]=$time[$l+1];
$time[$l+1]=$temptime;
$tempmyrow=$myrow[$l];
$myrow[$l]=$myrow[$l+1];
$myrow[$l+1]=$tempmyrow;
unset ($tempmyrow);}
else if ($kol[$l]==$kol[$l+1]) {
if ($time[$l]>$time[$l+1]) {
$tempkol=$kol[$l];
$kol[$l]=$kol[$l+1];
$kol[$l+1]=$tempkol;
$temptime=$time[$l];
$time[$l]=$time[$l+1];
$time[$l+1]=$temptime;
$tempmyrow=$myrow[$l];
$myrow[$l]=$myrow[$l+1];
$myrow[$l+1]=$tempmyrow;
unset ($tempmyrow);}}}}
echo "<table bgcolor='FFFFFF' width='814'
border='1'><tr><td>№</td><td align='center'
width='.(800/$zadachvalue+1).">Имя</td>";
for ($i=2;$i<2*$zadachvalue+2;$i+=2) {
```

```

echo "<td align='center'
width=".(800/$zadachvalue+1).">".($i/2)."</td>";}
echo "<td align='center'>V</td><td align='center'>T</td></tr>";
for ($k=0;$k<$j;$k++){
echo "<tr><td>". $k. "</td><td>". $myrow[$k][1]. "</td>";
for ($i=2;$i<2*$zadachvalue+2;$i+=2){
$subs = explode(':', $myrow[$k][$i+1]);
if (empty($myrow[$k][$i])==false)echo "<td
align='center'>". $myrow[$k][$i]. "<br>". $subs[0]. ":". $subs[1]. "</
td>";
else echo "<td align='center'></td>";}
echo "<td align='center'>". $kol[$k]. "</td><td
align='center'>". $ttime[$k]. "</td></tr>";}
echo "</table>";}
?>

```

Таким образом, разработанная система работает и способна создавать и проводить турниры, так как все требуемые компоненты уже созданы. Но системе всё ещё нужны доработки, а именно:

1. Разграничение прав администратора и участника.
2. Доработка интерфейса.
3. Увеличение числа проверяемых языков программирования.
4. Доработка таблицы лидеров (её заморозка за час до окончания турнира).
5. Расширение возможностей взаимодействия с турнирами (редактирование, удаление, быстрые правки).
6. Добавление возможности хранения логов всех решений пользователя.
7. Личный кабинет.
8. Правка мелких багов.

### Библиографический список

1. Олимпиады по программированию [Электронный ресурс] URL: [https://ru.wikipedia.org/wiki/Олимпиады\\_по\\_программированию](https://ru.wikipedia.org/wiki/Олимпиады_по_программированию) (дата обращения 16.12.2016)
2. Соревнования по программированию [Электронный ресурс] URL: <https://cs.hse.ru/olymp/Contests> (дата обращения 16.12.2016)
3. О системе Contester [Электронный ресурс] URL: <http://ulivt.ru:8080/ru/about> (дата обращения 16.12.2016)
4. Contester [Электронный ресурс] URL: <http://www.contester.ru/> (дата обращения 13.10.2016)
5. Долгушин Н.А., Оленькова М.Н. Использование системы contester для проведения олимпиад по программированию // Материалы VIII Международной студенческой электронной научной конференции «Студенческий научный форум» URL: <http://www.scienceforum.ru/2016/1728/21959> (дата обращения: 16.12.2016).



- 
6. Система CATS [Электронный ресурс] URL: <https://imcs.dvfu.ru/cats/> (дата обращения 16.12.2016)
  7. Система ejudge [Электронный ресурс] URL: <https://ejudge.ru/> (дата обращения 16.12.2016)