

КПО

Автоматизация тестирования

Лекция №10 (версия 1.0)

```
sent"/>  
fish.web.present
```

```
<!-- do not forg
```

```
oot}" else="$}{gfv
```

```
app.context-root]
```

```
resent">
```

```
b}"/>
```

Unit-тестирование

Модульное тестирование или юнит-тестирование (англ. unit testing) — процесс, позволяющий проверить на корректность отдельные модули исходного кода программы.

Идея состоит в том, чтобы писать тесты для каждой нетривиальной функции или метода. Это позволяет достаточно быстро проверить, не привело ли очередное изменение кода к регрессии, то есть к появлению ошибок в уже написанных и оттестированных местах программы, а также облегчает обнаружение и устранение таких ошибок.

Unit-тестирование

Хорошие юнит тесты:

- Автоматизированы
- Выполняются быстро: выполнение сотни или даже тысячи юнит-тестов должно занимать считанные секунды
- Написаны в рамках эффективного фреймворка
- Независимыми друг от друга
- Результат работы теста воспроизводим и повторяем
- Легко поддаются отладке
- Написаны автором кода
- Оставляют конфигурацию / окружение в неизменном состоянии
- Имеют высокое покрытие кода
- Поддаются хранению и сопровождению
- Тестируют как функциональность, так и аварийные ситуации
- Покрывают граничные значения входных параметров

Unit-тестирование

Юнит-тесты имеют и некоторые ограничения, о которых тоже нужно помнить:

- Юнит-тестирование не находит все ошибки в коде: оно показывает только наличие ошибок, а не их отсутствие
- У юнит-тестирования ограниченные рамки: это тестирование «белого ящика»
- Юнит-тестирование не покрывает сценарии и интеграцию между компонентами

Unit-тестирование

Одной из наиболее важных техник, которые позволяют создать хорошие юнит-тесты, является техника использование mock-объектов (mock objects). **Mock-объект - это симуляция реального объекта там, где использование реального объекта невозможно или нарушает принципы юнит-тестирования**, например когда реальный объект:

- предоставляет недетерминированные результаты (например текущее время или температуру)
- имеет трудно воспроизводимые состояния (например ошибка сети)
- медленный (например полная база данных или файл на диске)
- ещё не существует или может изменить поведение
- должен содержать информацию или методы, необходимые только для тестирования

Unit-тестирование

Для большинства популярных языков программирования высокого уровня существуют инструменты и библиотеки модульного тестирования. Некоторые из них:

Для Java

- JUnit JUnit.org
- TestNG testNG.org
- JavaTESK UniTESK.ru

NUnit [1] — для языков платформы .NET: C#, Visual Basic .NET и др.

Для C

- CUnit [cunit](http://cunit.com)
- CTESK UniTESK.ru
- cfix [cfix](http://cfix.com)
- API Sanity Autotest — для динамических C/C++ библиотек в Unix-подобных ОС.
- Для Objective-C
- OCUit [2]

Для C++

- CPPUnit [3]
- Boost Test [4]
- Google C++ Testing Framework [5]
- Symbian[6] — фреймворк для Symbian OS всех версий.
- API Sanity Autotest — для динамических C/C++ библиотек в Unix-подобных ОС.

Unit-тестирование

DUnit [7] — для Delphi

Для Perl

- Test [8]
- Test::Simple [9]
- Test::Unit [10]
- Test::Unit::Lite [11]

Для PHP

- SimpleTest [12]
- PHPUnit [13]

Для Python

- PyUnit [14]
- PyTest [15]
- Nose [16]

vbUnit [17] — Visual Basic

utPLSQL [18] — PL/SQL

Для T-SQL

TSQLUnit [19]

SPUnit [20]

Test::Unit [25] — для Ruby

JsUnit [26] — для JavaScript

```
sent"/>  
fish.web.present  
  
<!-- do not forg
```

```
oot}" else="$ {gfv
```

```
app.context-root]
```

```
resent">
```

```
b]"/>
```

Unit-тестирование

```
sent"/>
fish.web.present
<!-- do not forg
```

MyClass.php

```
<?php
class MyClass
{
    public function power($x, $y)
    {
        return $x ** $y;
    }
}
```

MyClassTest.php

```
<?php
require_once 'PHPUnit/Framework.php';
require_once 'MyClass.php';

class MyClassTest extends PHPUnit_Framework_TestCase {
    public function testPower()
    {
        $my = new MyClass();
        $this->assertEquals(8, $my->power(2, 3));
    }
}
```

```
$ phpunit MyClassTest
```

```
.
```

```
Time: 0 seconds
```

```
OK (1 test, 1 assertion)
```


Unit-тестирование

Базовые методы сравнения

assertTrue() / assertFalse()
assertEquals() / assertNotEquals()
assertGreaterThan()
assertGreaterThanOrEqual()
assertLessThan()
assertLessThanOrEqual()
assertNull() / assertNotNull()
assertType() / assertNotType()
assertSame() / assertNotSame()
assertRegExp() / assertNotRegExp()

Методы сравнения массивов

assertArrayHasKey() / assertArrayNotHasKey()
assertContains() / assertNotContains()
assertContainsOnly() / assertNotContainsOnly()

ООП специфичные методы

assertClassHasAttribute() / assertClassNotHasAttribute()
assertClassHasStaticAttribute() / assertClassNotHasStaticAttribute()
assertAttributeContains() / assertAttributeNotContains()
assertObjectHasAttribute() / assertObjectNotHasAttribute()
assertAttributeGreaterThan()
assertAttributeGreaterThanOrEqual()
assertAttributeLessThan()
assertAttributeLessThanOrEqual()

```
resent"/>  
fish.web.present  
  
<!-- do not forg
```

```
oot}" else="$}{gfv
```

```
app.context-root}
```

```
resent">
```

```
b}"/>
```

Unit-тестирование

```
2 <project def
3   <target r
4     <proj
5     <ava
6     <ava
7     <ava
8     <tem
9     <ech
```

```
sent"/>
fish.web.present
<!-- do not forg
```

Листинг 14. Простой тестовый класс JUnit 4

```
11
12 import java.util.regex.Matcher;
13 import java.util.regex.Pattern;
14
15 import org.junit.BeforeClass;
16 import org.junit.Test;
17 import static org.junit.Assert.assertTrue;
18
19 public class RegularExpressionTest {
20     private static String zipRegEx = "[0-9]{5}-[0-9]{4}";
21     private static Pattern pattern = Pattern.compile(zipRegEx);
22
23     @BeforeClass
24     public static void setUpBeforeClass() throws Exception {
25         pattern = Pattern.compile(zipRegEx);
26     }
27
28     @Test
29     public void verifyGoodZipCode() {
30         Matcher mtcher = this.pattern.matcher("12345-6789");
31         boolean isValid = mtcher.matches();
32         assertTrue("Pattern did not validate", isValid);
33     }
34 }
35
36 <copy
```

Листинг 15. Множество ошибок

```
[junit] Running test.com.acme.RegularExpressionTest
[junit] Tests run: 1, Failures: 1, Errors: 0, Time elapsed: 0.047 sec
[junit] Testsuite: test.com.acme.RegularExpressionTest
[junit] Tests run: 1, Failures: 1, Errors: 0, Time elapsed: 0.047 sec
[junit] Testcase: warning took 0.016 sec
[junit] FAILED
[junit] No tests found in test.com.acme.RegularExpressionTest
[junit] junit.framework.AssertionFailedError: No tests found in
test.com.acme.RegularExpressionTest
[junit] Test test.com.acme.RegularExpressionTest FAILED
```

Code Coverage

The screenshot illustrates the process of analyzing code coverage in Visual Studio. The 'Analyze Code Coverage' menu is open, showing options for 'Selected Tests' and 'All Tests'. The 'Code Coverage Results' window is open, displaying a table with the following data:

Hierarchy	Not Cov...	Not Covered (%...	Cov...
ctsoasm_MAIN50531 201...	44	80.00%	11
fabrikam.math.dll	7	50.00%	7
{ } Fabrikam.Math	7	50.00%	7

The code for the `SquareRoot` method is shown with the following annotations:

- `if (x < 0.0)`: Not covered
- `throw new ArgumentOutOfRangeException();`: Not covered
- `double estimate = x;`: Covered
- `double previousEstimate = -x;`: Covered
- `while (System.Math.Abs(estimate - previousEstimate) >...)`: Covered

The 'Code Coverage Results' window also shows a 'Turn on coloring' button and a '100%' zoom level.

```
sent"/>  
fish.web.present
```

```
<!-- do not forg
```

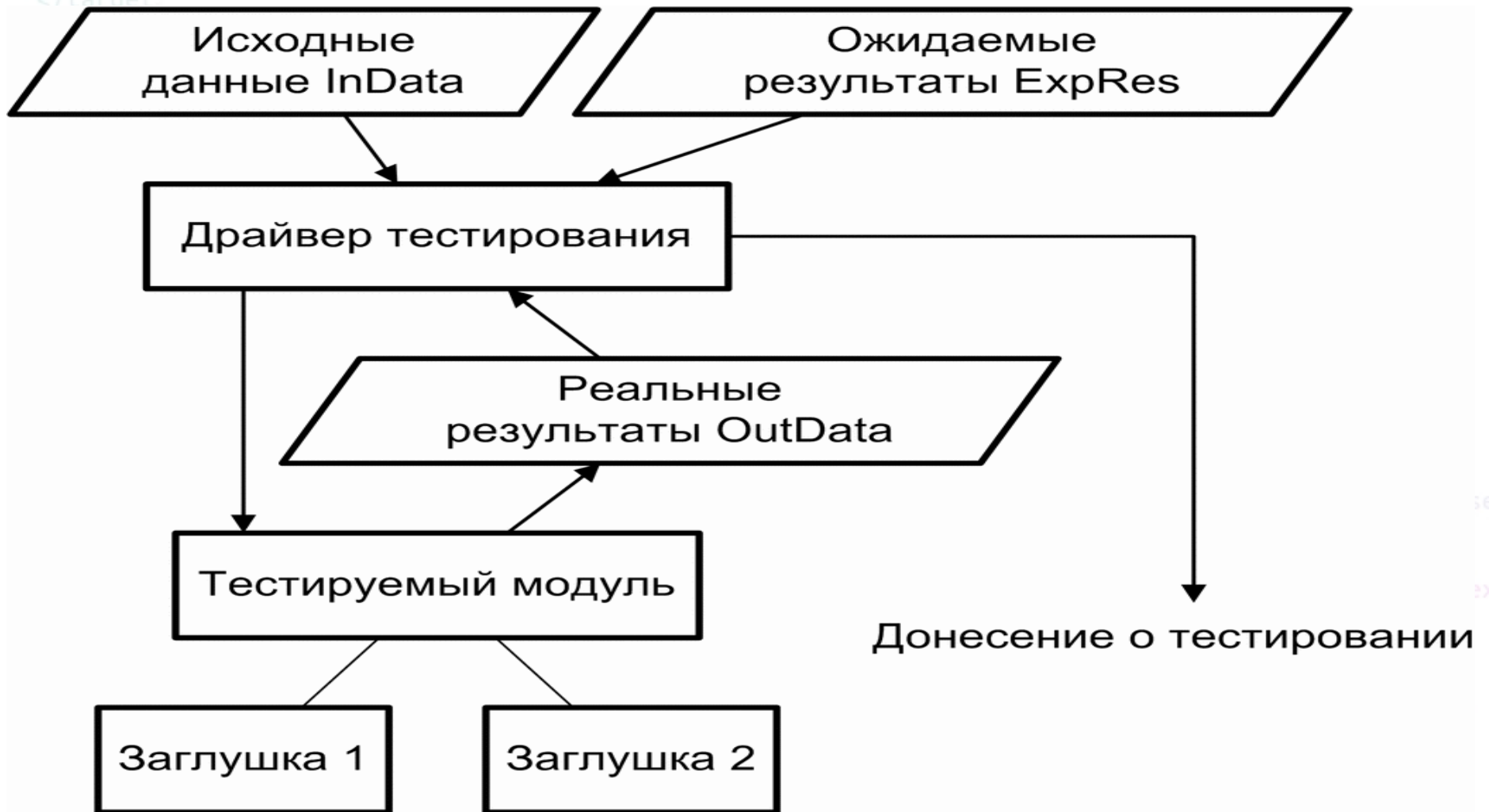
```
oot}" else="$gfv
```

```
app.context-root
```

```
resent">
```

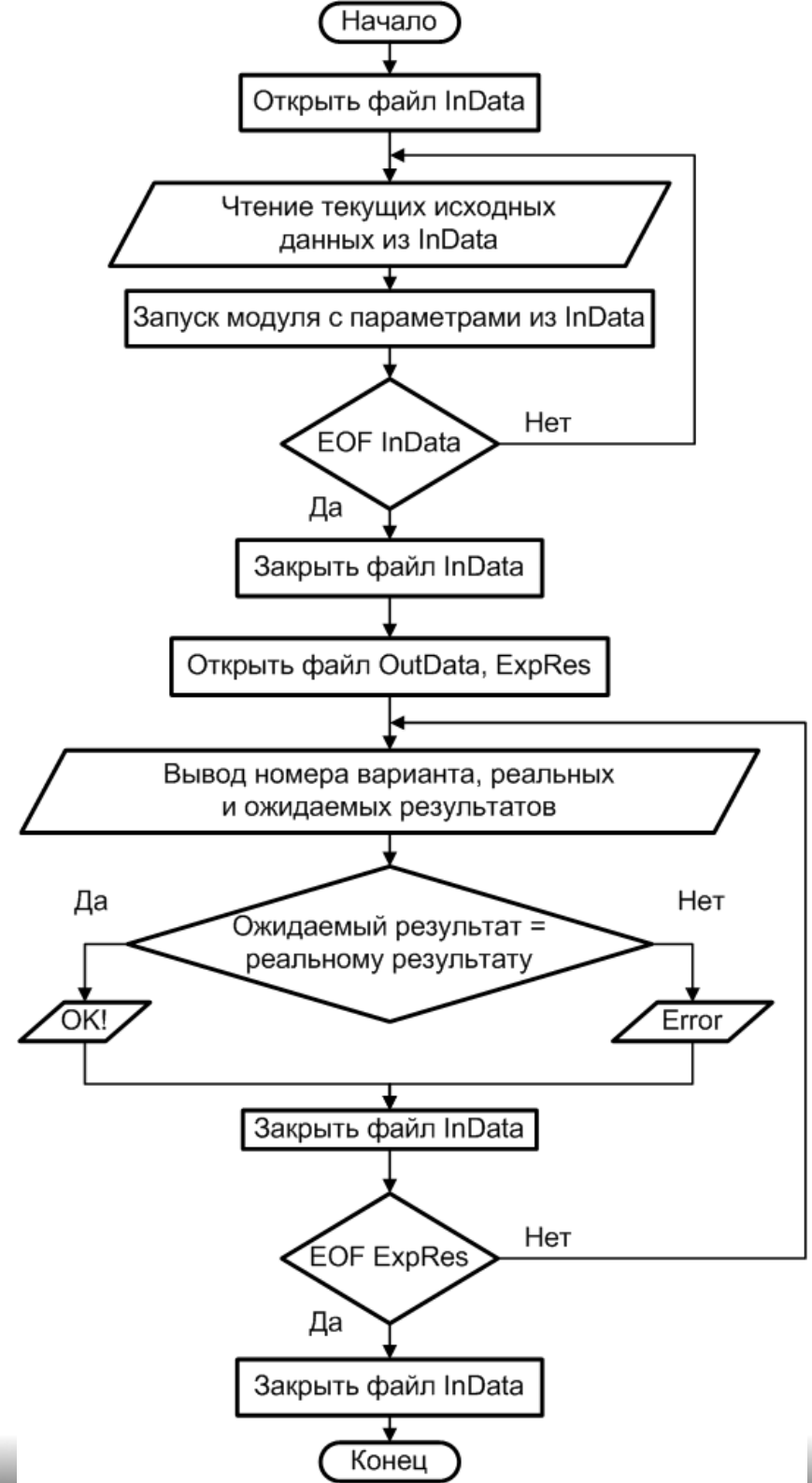
```
b}"/>
```

Драйвер тестирования



```
2 <project def
3   <target r
4     <proj
5     <ava
6     <ava
7     <ava
8     <temp
9     <ech
10  </target>
11
12  <target r
13    <temp
14    <copy
15    <!--
16    <repl
17      <
18    </repl
19    <repl
20      <
21    </repl
22    <
23  </repl
24  <xmlp
25  </xml
26  <dele
27  <cond
28    <
29  </con
30  <cond
31    <
32  </con
33 </target>
34 <target n
35   <temp
36   <copy
```

```
sent"/>
fish.web.present
<!-- do not forg
oot}" else="$gfv
app.context-root]
resent">
b]"/>
```

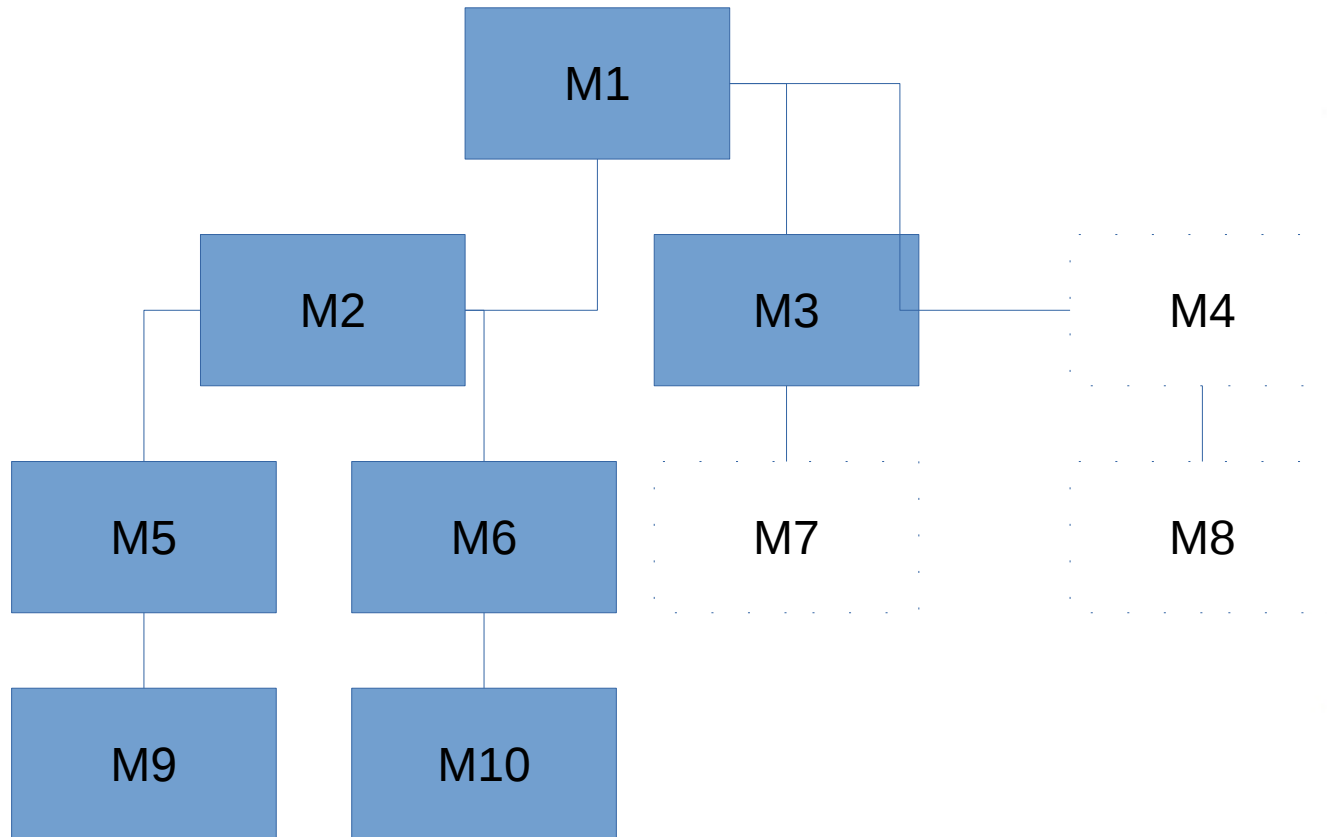


Тестирование интеграции

Категории ошибок интерфейса:

- Потеря данных при прохождении через интерфейс;
- Отсутствие в модуле необходимой ссылки;
- Негативное влияние одного модуля на другой;
- Подфункции при объединении не выполняют требуемую функцию;
- Неточности интеграции выходят за допустимый уровень;
- Проблемы при работе с глобальными структурами данных.

Нисходящее тестирование



```
sent"/>  
fish.web.present  
<!-- do not forg  
ot}" else="$gfv  
app.context-root  
resent">  
b]"/>
```

Типы заглушек

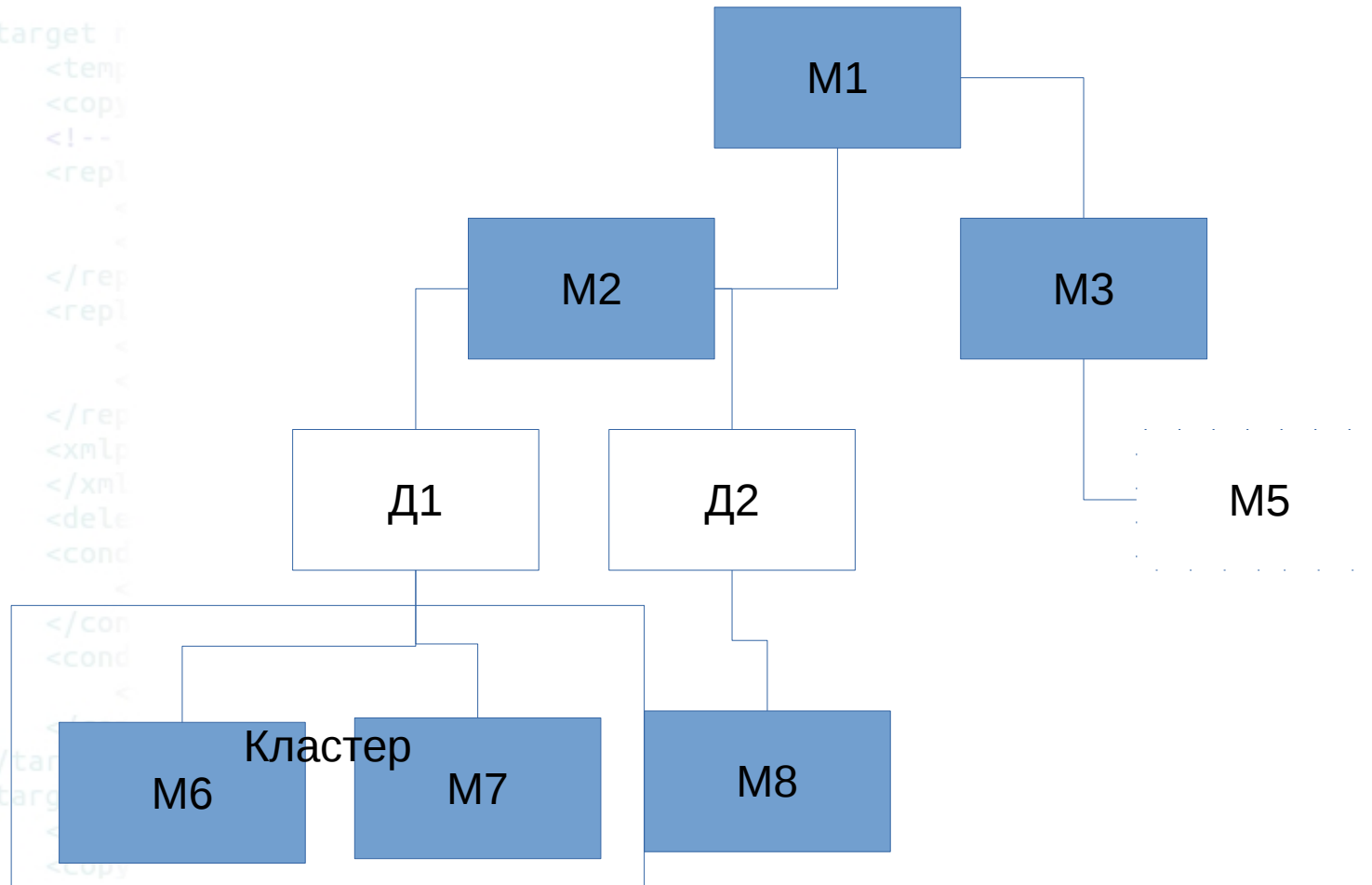
- заглушка A – трассирует сообщение;
- заглушка B – выводит входные параметры;
- заглушка C – возвращает значения из таблицы, заданного набора;
- заглушка D – возвращает результат из таблицы для конкретного входного значения.

```
resent"/>
fish.web.present

<!-- do not forg

oot}" else="$gfv
app.context-root]
resent">
b]"/>
```


Восходящее тестирование



```
sent"/>  
fish.web.present  
  
<!-- do not forg  
  
oot}" else="$gfv  
app.context-root"  
  
resent">  
b]"/>
```

Типы драйверов

- драйвер А – вызывает подчиненный модуль;
- драйвер В – посылает элемент данных (параметр) из внутренней таблицы;
- драйвер С – отображает параметр из подчиненного модуля;
- драйвер D – является комбинацией драйверов В и С.

```
sent"/>  
fish.web.present  
  
<!-- do not forg  
  
oot}" else="$gfv  
  
app.context-root]  
  
resent">  
  
b]"/>
```

TPC Transaction Processing Performance Council

Синтетические тесты на производительность СУБД. Целью тестов TPC является предоставление объективных данных о производительности системы. Для того, чтобы достичь этой цели, спецификации TPC требуют, чтобы тестировалась производительность систем, продуктов, технологий, отвечающих следующим требованиям:

- продукт должен быть доступен пользователям;
- продукт должен относиться к соответствующему сегменту рынка (обработка транзакций);
- при тестировании продукта должны быть обязательно воспроизведены действия некоторого числа пользователей этой системы.

```
2 <project def
3   <target r
4     <proj
5     <ava
6     <ava
7     <ava
8     <temp
9     <ech
10  </target>
11
12 <target r
13   <temp
14   <copy
15   <!--
16   <rep
17   <
18   <
19   </rep
20   <rep
21   <
22   <
23   </rep
24   <xmlp
25   </xml
26   <dele
27   <cond
28   <
29   </con
30   <cond
31   <
32   </con
33 </target>
34 <target n
35   <temp
36   <copy
```

Тестирование СИРП

Скрипт запущен на сервере **web-edu.iriit 12.05.2006 13:15**

SoapClient успешно создан

Запуск теста **t1soap.php**

- T1s.1 **пройден**
- T1s.2 **пройден**
- T1s.3 **не пройден**

Файл с результатами **./file_out/t1soap1147407317.out**

Запуск теста **t2soap.php**

- T2s.1 **пройден**
- T2s.2 **пройден**
- T2s.3 **пройден**

Файл с результатами **./file_out/t2soap1147407317.out**

Запуск теста **t3soap.php**

- T3s.1 **пройден**
- T3s.2 **пройден**
- T3s.3 **пройден**

Файл с результатами **./file_out/t3soap1147407318.out**

```
sent"/>
fish.web.present
```

```
<!-- do not forg
```

```
oot}" else="$gfv
```

```
app.context-root
```

```
resent">
```

```
b]"/>
```



Система тестирования функций

Система: SOAP для студентов

Список функций

Название	Тип	Система	Статус	Действие	Результат
NameStudGroup_PO	soap	SOAP для студентов	выкл.		Сбой
StudGroupList_PO	soap	SOAP для студентов	вкл.		Пройден
stud_PredmetName	soap	SOAP для студентов	вкл.		Пройден
stud_PredStat	soap	SOAP для студентов	вкл.		Не пройден
stud_PrepInfo	soap	SOAP для студентов	вкл.		Пройден
stud_SessionKurs	soap	SOAP для студентов	вкл.		Не пройден
stud_SpisokPred	soap	SOAP для студентов	вкл.		Пройден
stud_SpisokStud	soap	SOAP для студентов	вкл.		Не пройден
stud_StudGroupInfo	soap	SOAP для студентов	вкл.		Пройден
stud_StudInfo	soap	SOAP для студентов	вкл.		Пройден

SOAP для студентов

РиСПО © 2006-2007, К.Г. Колодий, v 1.2



Система тестирования функций

Система: SOAP для студентов

Список тестовых вариантов

stud_StudInfo	soap	SOAP для студентов	вкл.
---------------	------	--------------------	------

Описание: Функция stud_StudInfo(\$stud_id) выдает информацию для заданного студента. Входного параметр - stud_id(id студента). Результатом функции является строка содержащая элементы - stud_id(id студента), fname(фамилия), lname(имя), mname(отчество) и foto.

Вход	Выход	Статус	Действия
stud_id	res		
Описание: T1			
"null"	Array ("stud_id"=>"2236", "fname"=>"Мирошниченко", "lname"=>"Евгения", "mname"=>"Васильевна", "foto"=>"пусто")	вкл.	
Описание: T2			
-1	Array ("stud_id"=>"2236", "fname"=>"Мирошниченко", "lname"=>"Евгения", "mname"=>"Васильевна", "foto"=>"пусто")	вкл.	
Описание: T3			
9730	Array ("stud_id"=>"2236", "fname"=>"Мирошниченко", "lname"=>"Евгения", "mname"=>"Васильевна", "foto"=>"пусто")	вкл.	

Описание

```
2 <project def
3   <target r
4     <proj
5     <aval
6     <aval
7     <aval
8     <temp
9     <ech
10  </target>
11
12  <target r
13    <temp
14    <copy
15    <!--
16    <repl
17    <
18    <
19    </rep
20    <repl
21    <
22    <
23    </rep
24    <xmlp
25    </xml
26    <dele
27    <cond
28    <
29    </con
30    <cond
31    <
32    </con
33  </target>
34  <target n
35    <temp
36    <copy
```

```
sent"/>
fish.web.present
```

```
<!-- do not forg
```



```
oot}" else="{gfv
```

```
app.context-root]
```

```
resent">
```

```
b}"/>
```

```

2 <project def
3   <target
4     <proj
5     <ava
6     <ava

```

```

sent"/>
fish.web.present

```

Код ошибки	▲ Дата ▼	Ссылка	Короткое описание	▲ Статус ▼	▲ Класс ▼	создан	▲ Reported from ▼	назначен
2007-01-471	9/1/2007	Сервер для разработчиков	eis_VPrikazAdd (173)	Исправлена	простая	К. Колодий	Система тестирования SOAP	SOAP
2007-01-498	24/1/2007	Деканат 1.25	Форма оплаты	Исправлена	предложение	В. Федоров	В. Федоров	Е. Арбатский
2007-02-520	2/2/2007	Деканат 1.25	Дубликаты	Исправлена	сложная	В. Федоров	В. Федоров	Е. Арбатский
2007-02-525	5/2/2007	Интерфейс	Нет данных по отчисленным	Исправлена	простая	К. Колодий	К. Колодий	Е. Арбатский
2007-02-533	6/2/2007	Сервер для РиСПО	функция изменения статуса студента	Исправлена	простая	К. Колодий	К. Колодий	Е. Арбатский
2007-02-637	13/2/2007	Деканат 1.25	Список с паспортами	Исправлена	предложение	Е. Арбатский	Е. Арбатский	В. Федоров
2007-03-655	13/3/2007	Деканат 1.25	Группы в архиве	Исправлена	опасная	Е. Арбатский	Е. Арбатский	Е. Арбатский
2007-03-669	19/3/2007	Дипломное проектирование 2.0	Неверная форма оплаты	Исправлена	простая	Е. Арбатский	Е. Арбатский	Е. Арбатский
2007-03-670	19/3/2007	Дипломное проектирование 2.0	Нет студентов	Исправлена	сложная	Е. Арбатский	Е. Арбатский	Е. Арбатский
2007-03-671	19/3/2007	Дипломное проектирование 2.0	Группы в дереве	Исправлена	опасная	Е. Арбатский	Е. Арбатский	В. Федоров
2007-04-721	9/4/2007	Приложения к диплому	Не тот план	Исправлена	опасная	Е. Арбатский	Е. Арбатский	В. Федоров
2007-04-722	9/4/2007	Дипломное проектирование 2.0	Списки преподавателей	Исправлена	опасная	Е. Арбатский	Е. Арбатский	В. Федоров
2007-04-724	10/4/2007	Деканат 1.25	Двойки в экзам. карточках	Исправлена	предложение	Е. Арбатский	Е. Арбатский	В. Федоров
2007-01-468	1/1/2007	Сервер для разработчиков	eis_VPrikazAdd (173)	Закрота	простая	К. Колодий	Система тестирования SOAP	Система тестирования SOAP
2007-01-469	1/1/2007	Сервер для РиСПО	getnweek (112)	Закрота	простая	К. Колодий	Система тестирования SOAP	Система тестирования SOAP


```

2 <project def
3   <target r
4     <proj
5     <ava
6     <ava
7     <ava
8     <temp
9     <ech
10  </target>
11
12 <target r
13   <temp
14   <copy
15   <!--
16   <rep
17   <
18   <
19   </rep
20   <rep
21   <
22   <
23   </rep
24   <xmlp
25   </xml
26   <dele
27   <cond
28   <
29   </con
30   <cond
31   <
32   </con
33 </target>
34 <target n
35   <temp
36   <copy

```

Показано Ошибка 2007-04-721 ✎ 🗑

создан Евгений Викторович Арбатский в 09/04/2007 14:27 IRKST

Код ошибки	2007-04-721	
Задача	Приложения к диплому	
Статус	Исправлена	Класс опасная
исправлен	Вячеслав Викторович Федоров	в 20/04/2007 12:05 IRKST (14/4/2007)
Reported from	Евгений Викторович Арбатский	(Reported Bugs Overview)
How detected/reported	Desktop	
Короткое описание	Не тот план	
История	<p>09/04/2007 14:27 IRKST Евгений Викторович Арбатский</p> <p>В тестовом деканате создается план для ИС-02-1, открывается на редактирование и показывается план для специальности ПО.</p> <p>Следует переделать редактирование плана!</p> <hr/> <p>09/04/2007 14:27 IRKST Евгений Викторович Арбатский</p> <p>назначен: Евгений Викторович Арбатский → Вячеслав Викторович Федоров</p> <hr/> <p>17/04/2007 09:42 IRKST Вячеслав Викторович Федоров</p> <p>Были перпутаны названия групп. Исправлено. Разобраться с количеством часов в плане и приложения для каждого студента.</p> <hr/> <p>17/04/2007 15:32 IRKST Евгений Викторович Арбатский</p> <p>надо еще разобраться с часами. Обсудил вопрос с Анной Владимировной (17.04) - она сказала что посмотрит. Надо будет у нее узнать результаты.</p> <hr/> <p>20/04/2007 12:03 IRKST Евгений Викторович Арбатский</p> <p>не хватало циклов для предметов</p> <hr/> <p>20/04/2007 12:05 IRKST Евгений Викторович Арбатский</p> <p>Статус: Открытая → Исправлена</p>	

```
2 <project def
3   <target r
4     <proj
5     <ava
6     <ava
7     <ava
8     <temp
9     <ech
10  </target>
11
12  <target r
13    <temp
14    <copy
15    <!--
16    <repl
17      <
18    </repl
19  </repl
20  <repl
21    <
22  </repl
23  </repl
24  <xmlp
25  </xml
26  <dele
27  <cond
28    <
29  </con
30  <cond
31    <
32  </con
33 </target>
34 <target n
35   <temp
36   <copy
```

```
sent"/>
fish.web.present
<!-- do not forg
```

Показано Ошибка 2007-01-502

создан Константин Колодий в 29/01/2007 06:24 IRKT

Код ошибки	2007-01-502
Задача	Сервер для РИСПО
Статус	Закрота Класс простая
закрывает	Система тестирования SOAP в 05/02/2007 06:28 IRKT (--/--/----
Reported from	Система тестирования SOAP (Reported Bugs Overview)
How detected/reported	Miscellaneous
Короткое описание	getnweek (112)
История	29/01/2007 06:24 IRKT Константин Колодий

Не пройден тест (230).
 Значения входных переменных:
 day = "null"
 month = 9
 year = 2007

Ожидаемые значения выходных переменных:
 res = "null"

Реальные значения выходных переменных:
 res = 1

```
oot}" else="$gfv
app.context-root}
```

```
resent">
b}"/>
```

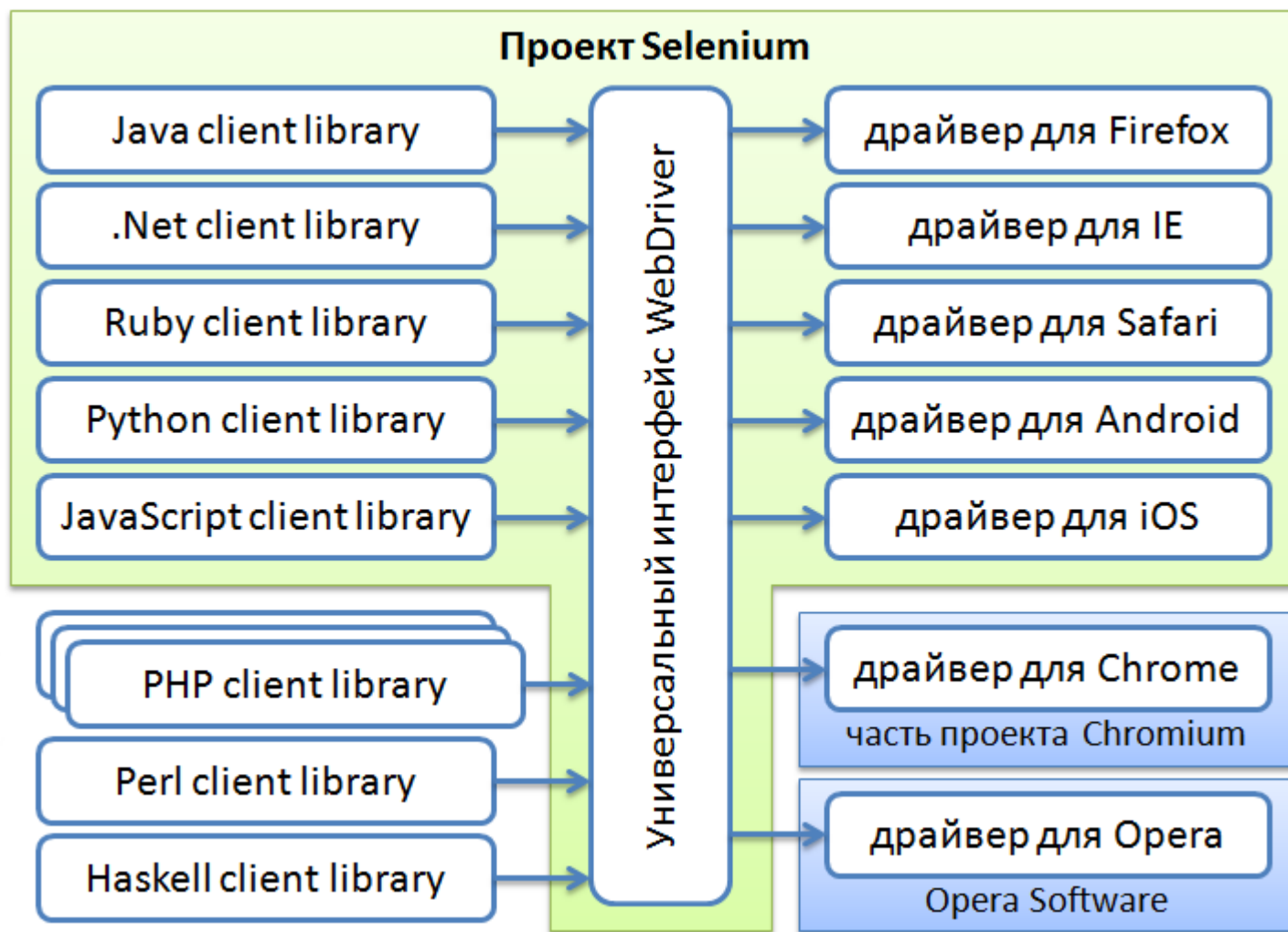
Selenium

Selenium — это инструмент для тестирования Web-приложений.

Selenium - это Java-приложение, которое может анализировать файлы определенной структуры для того, чтобы находить в них команды для манипуляции браузером и команды для выполнения определенных действий и проверок.

Selenium поддерживается Microsoft Internet Explorer, Google Chrome, Mozilla Suite и Mozilla Firefox для Microsoft Windows, Linux и Apple Macintosh.

Selenium



```
sent"/>  
fish.web.present  
<!-- do not forg
```

```
oot}" else="$gfv  
app.context-root"  
resent">  
b]"/>
```

Selenium

```
sent"/>  
fish.web.present  
<!-- do not for
```

Downloads - Mozilla Firefox

File Edit View History Bookmarks Tools Help

- Web Search Ctrl+K
- Downloads Ctrl+J
- Add-ons
- Error Console Ctrl+Shift+J
- AdBlock Plus Preferences... Ctrl+Shift+E
- Page Info
- Selenium IDE**
- Start Private Browsing Ctrl+Shift+P
- Clear Recent History... Ctrl+Shift+Del
- Options...

Selenium
Web application testing system

Selenium Downloads
Previous Releases
Nightly Releases
Source Code
Maven Repository Info

Project	Release Date	Version	
Selenium Core	June 10, 2009	1.0.1	Download Changelog
Selenium IDE	Nov 5, 2010	1.0.8	Download Release Notes
Selenium RC	February 23, 2010	1.0.3	Download
Selenium Grid	April 8, 2010	1.0.6	Download (zip)
CubicTest	Nov 10, 2008	1.8.11	Download Changelog
Bromine	July 25, 2010	3.0 RC 2	Download
Selenium 2	April 22, 2010	2.0 alpha 5	Download

Plugins for Selenium IDE

Name	Author	License	Release
			September 2010

Done

Selenium IDE 1.0.8

File Edit Options Help

Base URL

Fast Slow

Table Source

Command	Target	Value

Command

Target Find

Value

Log Reference UI-Element Rollup Info Clear

TestComplete

```
2 <project def  
3   <target r  
4     <proj  
5     <aval  
6     <aval  
7     <aval  
8     <temp  
9     <ech  
10  </target>  
11  
12  <target r  
13    <temp  
14    <copy  
15    <!--  
16    <repl  
17    <  
18    <  
19    </repl  
20    <repl  
21    <  
22    <  
23    </repl  
24    <xmlp  
25    </xml  
26    <dele  
27    <cond  
28    <  
29    </con  
30    <cond  
31    <  
32    </con  
33  </target>  
34  <target n  
35    <temp  
36    <copy
```

```
sent"/>  
fish.web.present
```

```
<!-- do not forg
```



```
oot}" else="$ {gfv
```

```
app.context-root]
```

```
resent">
```

```
b]"/>
```

IBM Rational TestManager

The screenshot displays the IBM Rational TestManager interface. The main window is titled 'test_sirp - Rational Robot - [menu]'. It features a menu bar (File, Edit, View, Rec) and a toolbar with icons for GUI, VU, and other test actions. The central area shows a tree view of test events, and a table on the right lists the results of these events.

Event Type	Result	Date & Time
Computer Start (Temporary S...	Fail	07.05.2006 12:53:14
TestCase Start	Pass	07.05.2006 12:53:17
Script Start (menu)	Pass	07.05.2006 12:53:17
Script End (menu)	Pass	07.05.2006 12:53:35
TestCase End	Pass	07.05.2006 12:53:35
TestCase Start	Fail	07.05.2006 12:53:35
Script Start (vizit)	Fail	07.05.2006 12:53:35
Script Command ...	Fail	07.05.2006 12:54:24
Script End (vizit)	Fail	07.05.2006 12:54:24
Message	Informational	07.05.2006 12:54:24
TestCase End	Fail	07.05.2006 12:54:24
TestCase Start	Fail	07.05.2006 12:54:24
Script Start (otmetka)	Fail	07.05.2006 12:54:24
Playback Warning	Warning	07.05.2006 12:54:55
Script Command ...	Fail	07.05.2006 12:55:42
Script End (otme...	Fail	07.05.2006 12:55:42
Message	Informational	07.05.2006 12:55:42
TestCase End	Fail	07.05.2006 12:55:42
TestCase Start	Pass	07.05.2006 12:55:42
Script Start (predmet)	Pass	07.05.2006 12:55:42
Script End (pred...	Pass	07.05.2006 12:55:48
TestCase End	Pass	07.05.2006 12:55:48
Computer End	Fail	07.05.2006 12:55:56
Suite End (Temporary Suite 1)	Fail	07.05.2006 12:55:59

The bottom status bar shows 'Build Console' and the user 'admin'.

```
sent"/>  
fish.web.present
```

```
<!-- do not for
```

```
'sirp/', "W
```

```
1,7"  
sirp{ENTER}
```

```
ивидуальной  
ивидуальной
```

```
oot)" else="$gfv
```

```
app.context-root
```

```
resent">
```

```
b)"/>
```

IBM Rational TestManager

The screenshot displays the IBM Rational TestManager interface. The window title is "seminar - Rational TestManager - [Test Log - TimerCalcDP32]". The menu bar includes File, Edit, View, Reports, Tools, Window, and Help. The main area shows the "Test Log" for "Timer Perf" under "Build 1" and "Log Folder: Default". The interface includes a toolbar with various icons for file operations and testing. The main pane displays a table of test events.

Event Type	Result	Date & Time
Log Message (30/10=3)	Pass	5/22/2003 12:32:44
Timer End	Pass	5/22/2003 12:32:45
Timer Start	Pass	5/22/2003 12:32:45
Log Message (34/34=1)	Pass	5/22/2003 12:32:47
Timer End	Pass	5/22/2003 12:32:48
Timer Start	Pass	5/22/2003 12:32:48
Log Message (38/19=2)	Pass	5/22/2003 12:32:50
Timer End	Pass	5/22/2003 12:32:51
Purify: W PAR: Global/LocalHandle(0x0) arg #1 (pMem) does not p...	Warning	5/22/2003 12:32:51
Purify: I Terminating thread 0x70c	Informational	5/22/2003 12:32:51
Purify: I Searching for all memory leaks...	Informational	5/22/2003 12:32:54
Purify: W MLK: Memory leak of 32 bytes from 1 block allocated in L...	Warning	5/22/2003 12:32:55
Purify: W MLK: Memory leak of 64 bytes from 1 block allocated in S...	Warning	5/22/2003 12:32:56
Purify: W MLK: Memory leak of 8 bytes from 1 block allocated in Lo...	Warning	5/22/2003 12:32:57
Purify: W MLK: Memory leak of 28 bytes from 2 blocks allocated in ...	Warning	5/22/2003 12:32:58
Purify: I MPK: Potential memory leak of 20 bytes from 1 block allocat...	Informational	5/22/2003 12:32:59
Purify: I MPK: Potential memory leak of 60 bytes from 2 blocks alloc...	Informational	5/22/2003 12:33:00
Purify: W MLK: Memory leak of 68 bytes from 3 blocks allocated in ...	Warning	5/22/2003 12:33:01
Purify: W MLK: Memory leak of 92 bytes from 4 blocks allocated in ...	Warning	5/22/2003 12:33:02
Purify: I MPK: Potential memory leak of 208 bytes from 3 blocks allo...	Informational	5/22/2003 12:33:03
Purify: W MLK: Memory leak of 124 bytes from 5 blocks allocated in...	Warning	5/22/2003 12:33:04
Purify: W MLK: Memory leak of 128 bytes from 6 blocks allocated in...	Warning	5/22/2003 12:33:06
Purify: W MLK: Memory leak of 148 bytes from 1 block allocated in ...	Warning	5/22/2003 12:33:07
Purify: W MLK: Memory leak of 136 bytes from 1 block allocated in ...	Warning	5/22/2003 12:33:08
Purify: I MPK: Potential memory leak of 33 bytes from 1 block allocat...	Informational	5/22/2003 12:33:09
Purify: I Exiting with code 0 (0x00000000)	Informational	5/22/2003 12:33:10
Purify: I Program terminated at 05/22/2003 12:32:55	Informational	5/22/2003 12:33:11