

# ТЕХНОЛОГИИ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

УДК 159.9+004.42

## ОСОБЕННОСТИ РАЗРАБОТКИ ТЕСТОВ РАБОЧЕЙ ПАМЯТИ В ИНТЕРНЕТЕ

**В.К. Войтов, М.А. Сафонов, Р.Ф. Соколов**

Рассматриваются особенности разработки тестов рабочей памяти, ориентированных на Интернет.

---

The features of development of working memory tests to be implemented at the Internet are under consideration.

---

### 1. ВВЕДЕНИЕ

Память – одна из психических функций и видов умственной деятельности, предназначенная сохранять, накапливать и воспроизводить информацию.

Выделяют несколько видов памяти:

- долговременная память – способность к хранению информации (долговременную память о событиях называют эпизодической памятью, а память о фактах – семантической);
- рабочая память – это способность удерживать информацию и манипулировать ею в течение короткого времени (рабочая память помогает контролировать внимание и не отвлекаться при решении задач);
- кратковременную память некоторые исследователи отождествляют с рабочей памятью (но есть и те, кто разделяет эти два вида памяти).

Рабочая память выполняет оперативные задачи, требующие манипуляции с информацией. Для кратковременной памяти характерен относительно короткий период хранения информации, которая теряется после некоторого времени или из-за поступления новой информации. Долговременную память можно сравнить с жёстким диском, а рабочую память – с оперативной памятью.

Долговременная и рабочая память опираются на разные механизмы: в основе долговременной памяти лежат биохимические и клеточные процессы, в основе рабочей – психические.

По мнению ряда исследований, объем рабочей памяти определяет наш интеллектуальный уровень (Хайнц-Мартин Зюсс). Связь между объемом рабочей памяти и общим уровнем интеллекта прослеживается на уровне коэффициента корреляции, равного 0,6-0,8.

В современном мире всё шире и шире распространяются web технологии. Всё большее число людей становятся обладателями быстрого доступа в Интернет. По результатам исследования компании Pingdom на 2012 год, число пользователей Интернет приближалось 2.5 миллиарда человек в мире. В России это – 68 млн. человек. В странах СНГ – 93 млн. В

Европе – 519 млн. В Азии - 1.1 млрд пользователей. Таким образом, Интернет становится всё более и более эффективным средством для проведения Интернет тестирований. В настоящее время подобные тесты применяются для приёма на работу или индивидуальной диагностики памяти.

Программная реализация теста должна быть пригодная для работы в современных браузерах и опираться на применение современных компьютерных и web технологий. При этом некоторые программные инструменты и средства приходится разрабатывать с нуля, учитывая специфику и оригинальность тестов.

В этой статье рассмотрены функциональные и технические особенности разработки тестов рабочей памяти. В первой версии системы были реализованы следующие тесты: Reading Span, Operation Span, Symmetry Span, Running Span, Simple Spans, Visual WM, Verbal MUT, Visual MUT, n-Back.

Здесь не рассматривается реализация какого-то конкретного теста, а общие особенности их реализации. Во всех тестах тестируемому (пользователю) предъявляется последовательность блоков из фиксированного числа проб. В пробах каждого следующего блока число предъявляемых стимулов для запоминания увеличивается, поэтому сложность проб увеличивается. В качестве стимулов выступают слова, символы, таблицы, рисунки. В каждом тесте набор стимулов разный. Предъявление стимула занимает определённое время. Между предъявлениями очередных стимулов также должен выдерживаться некоторый интервал времени. В результате тестирования обычно определяется три величины:

- Количество правильных ответов;
- Количество проб, выполненных без неправильных ответов;
- Максимальная длина ряда целевых стимулов, при которой испытуемый полностью выполнил соответствующий блок.

Ниже на Рис. 1 приведён пример парных стимулов из теста **Symmetry Span**.



**Рис. 1. Пример парных стимулов из теста Symmetry Span. Сначала выдаётся стимул а), затем б).**

## 2. АРХИТЕКТУРА СИСТЕМЫ

Система тестирования состоит из набора тестов, подсистем регистрации, административная, записи в базу данных и выдачи отчетов. Тесты и подсистемы выполнены в виде отдельных файлов на РНР.

Пользователи системы делятся на три вида:

- обычный пользователь (испытуемый);
- администратор системы;
- администратор-психолог.

Обычный пользователь не имеет дополнительных привилегий. Он регистрируется, авторизуется и проходит выбранный тест.

Администратор системы – привилегированный пользователь системы. Он может создать или удалить базу данных и её таблицы, изменить некоторые параметры настройки системы. Администратор системы не может изменить полученные после тестирования данные.

Администратор-психолог – привилегированный пользователь системы. Привилегии назначаются администратором системы. Администратор-психолог имеет доступ к панели администратора-психолога, где он может просмотреть данные, касающиеся проведённого тестирования. Он может просмотреть список зарегистрированных пользователей, удалить некоторых пользователей. Администратор-психолог имеет возможность экспортировать полученные после тестирования данные в виде Excel-файлов.

Система состоит из следующих логических модулей:

- модуль регистрации и авторизации;
- модуль выбора теста;
- модуль администратора системы;
- модуль администратора-психолога;
- модуль записи результатов тестирования в базу данных;
- модуль выдачи отчетов;
- модуль отображения результатов тестирования;
- модуль тестирования.

На Рис. 2 приведена общая схема функционирования системы тестирования на уровне логических модулей. Модули реализованы с помощью средств PHP и JavaScript.

#### **Модуль регистрации и авторизации**

При входе в систему тестирования, пользователю предлагается авторизоваться в системе, введя свои регистрационные данные (используемые ранее) и нажав кнопку «Войти» или же пройти стандартную процедуру регистрации, нажав кнопку «Регистрация». Также имеется возможность указать группу, к которой принадлежит пользователь (например, служащий или учащийся). При регистрации информация о пользователе заносится в базу данных.

#### **Модуль выбора теста и его инициализация**

При корректной авторизации обычный пользователь попадает в меню выбора теста. При наведении указателя мыши на какой-либо тест, пункт меню раскрывается и в нем отображается краткое описание теста. При желании пройти необходимый тест, пользователю нужно нажать на кнопку «НАЧАТЬ ТЕСТИРОВАНИЕ» рядом с названием теста и он будет перенаправлен на страницу для прохождения выбранного теста. Если же пользователь не хочет участвовать в тестировании или закончил тестирование, то он может воспользоваться кнопкой «ВЫХОД» вверху экрана под приветствием, чтобы завершить сеанс. Информация о завершении сеанса будет занесена в базу данных.

#### **Модуль администратора системы**

Для администратора системы при входе в систему вызывается “Модуль администратора системы”. При этом отображается специальная панель, с помощью которой администратор может создать новую базу данных, удалить старую базу данных. Администратор имеет возможность просмотреть список пользователей и удалить некоторых из них. Он также может экспортировать базу данных в Excel. Администратор системы не имеет доступа к функциям администратора-психолога.

#### **Модуль выдачи отчётов**

После тестирования администратор-психолог может вызвать этот модуль для выполнения статистической обработки результатов тестирования и составления отчётов. Для оценки памяти тестируемого используются сохранённые в базе данных значения общего числа правильных ответов, число проб, выполненных без неправильных ответов и максимальная дли-

на ряда целевых стимулов, при которой испытуемый полностью выполнил соответствующий блок. Кроме того, предполагается учитывать время реакции испытуемого.

При статистической обработке результатов тестирования для расчета величин памяти предполагается использовать концепцию Г. Раша [1,2,3] (<http://it-fat.mgppu.ru>).

### Модуль рейтинга тестов

Для увеличения интереса пользователей к прохождению тестов введены таблицы рейтингов, которые отражают статистику тестирования. Рейтинги строятся с помощью SQL-запросов к базе данных. Поддерживается два типа рейтингов:

- рейтинг первых пяти тестов по популярности их использования среди пользователей;
- рейтинг пользователей (для каждого теста выдаются списки пользователей, занявших первые пять мест).

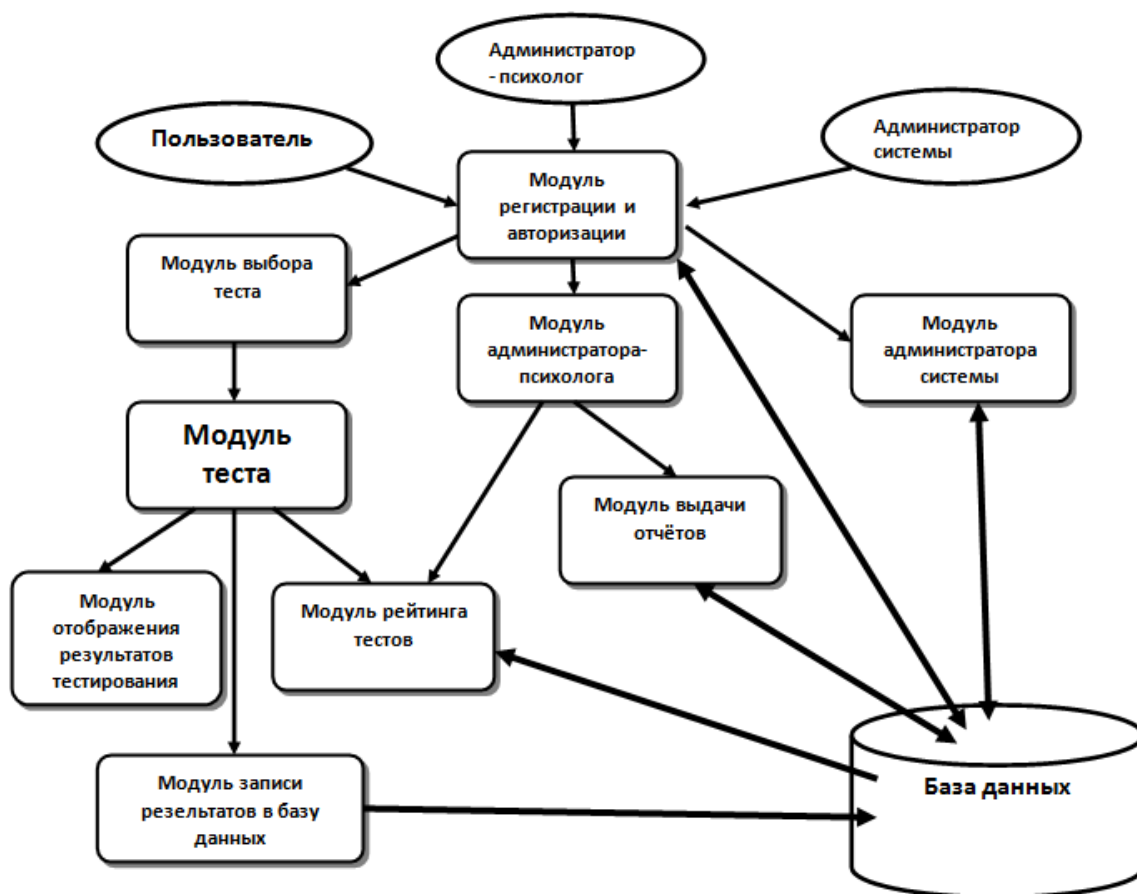


Рис. 2. Общая схема функционирования системы тестирования.

## 3. ОСОБЕННОСТИ РЕАЛИЗАЦИИ НЕКОТОРЫХ ФУНКЦИОНАЛЬНЫХ КОМПОНЕНТОВ

### 3.1. Отображение объектов

Диалог с пользователем и промежуточная обработка данных во время прохождения конкретного теста должны выполняться в локальном компьютере пользователя. Иначе из-за временных задержек при передаче данных в сети результаты тестирования будут некорректными. Поэтому лучше всего во время тестирования размещать на одном окне все нужные объекты, делая некоторые из них на время невидимыми, а другие видимыми.

Средства, позволяющие сделать объекты видимыми / невидимыми (скрытие отображения объектов):

- установка атрибута display;
- сдвиг объекта за границы окна;
- создание новых / удаление старых объектов;
- обращение к специальным программам.

Каждое из перечисленных здесь средства имеет свои преимущества и недостатки.

#### 1) установка атрибута display

Ниже приведён пример функции `dovis`, используемой для отображения объектов.

В функцию `dovis` передаётся идентификатор объекта `id` и флажок `v`. Если `v==1`, объекта становится видимым. В противном случае объект делается невидимым.

```
function dovis(v,id) {  
  var objid= document.getElementById(id);  
  if (v==1) { objid.style.display= "block";}  
  else {objid.style.display= "none";}  
}
```

#### 2) сдвиг объекта за границы окна

Ниже приведён пример функции `Move`, используемой для горизонтального сдвига всех объектов (тегов), определённых в теге `<DIV>`.

```
function Move(nid,how)  
{  var obj = document.all[nid].style;  
  obj.posLeft = how;  
}
```

...

```
Move("move",100)
```

...

```
<DIV id=move style="position:absolute; top:80; left:0; width:125">  
  <p> Тест № 4 </p>  
</DIV>
```

#### 3) создание новых / удаление старых объектов

С помощью методов `createElement()` и `removeChild()` можно создавать новые и удалять старые объекты. Это занимает дополнительное время и не всегда надёжно функционирует.

#### 4) обращение к специальным программам

Например, вызов `flash`-программы (файл с расширением `swf`) из `JavaScript`. Такая программа (пишется на языке `ActionScript`) обладает мощными средствами для манипулирования объектами. С помощью этих средств, например, реализован тест определения интеллекта Холлинга [4]: <http://it-fat.mgppu.ru/>

## 3.2. Структура базы данных

Система тестирования должна поддерживать интерфейс с пользователем и базу данных. База используется для хранения результатов тестирования и информации о зарегистрированных пользователях. База данных ведётся на сервере.

Для регистрации пользователей в базе данных поддерживается таблица `persons`. Так как пользователи могут несколько раз проходить один и тот же тест, в базе данных создаётся таблица `seans`, в которую заносится информация о каждом сеансе пользователя. Результаты

тестирования заносится в таблицу tests. Описания тестов находятся в таблице descrtest. На Рис. 3 приведена схема базы данных.

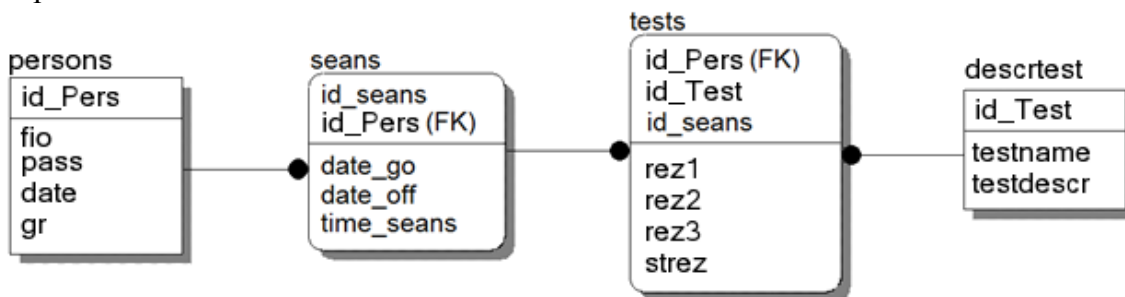


Рис 3. Схема базы данных.

### Таблица пользователей “persons”

В этой таблице хранится информация о регистрационных данных пользователя – логин, пароль, дата и время регистрации, группа. В таблице находятся соответствующие поля fio, pass, date, gr. А также, поле id\_Pers, которое является первичным ключом и идентификатором каждого пользователя.

### Таблица сеансов “seans”

В этой таблице хранятся данные о сеансах пользователя.

В поля date\_go, date\_off, time\_seans вносятся дата начала сеанса, дата завершения сеанса и время сеанса соответственно.

### Таблица результатов тестирования “tests”

В таблицу результатов вносятся три параметра характеризующие результаты тестирования – это rez1, rez2 и rez3. Они соответственно определяют три величины:

- количество правильных ответов;
- количество проб, выполненных без неправильных ответов;
- максимальную длину ряда целевых стимулов, при которой испытуемый полностью выполнил соответствующий блок.

В поле strez хранятся важные промежуточные результаты тестирования.

### Таблица описания теста “descrtest”

В данной таблице содержится подробное описание теста.

## 3.3. Работа с таймером.

Во время тестирования пользователю через определённые промежутки времени предъявляются различные объекты (стимулы). Пользователь должен отвечать на задаваемые в тесте вопросы в течение определённых интервалов времени. Тесты состоят из строгой последовательности этапов. Каждому из этапов выделяется определённый интервал времени. Время выполнения этапов не должно пересекаться. В противном случае возможно нарушение динамики тестирования и занесение неправильных результатов.

Для реализации правильного последовательного функционирования теста и обработки данных можно использовать:

- функции setTimeout и setInterval для установки временного интервала, после завершения которого следует вызвать требуемую функцию;
- функции clearTimeout и clearInterval для принудительного завершения действий функций setTimeout и setInterval .

В интернет на сайте <http://learn.javascript.ru/settimeout-setinterval> содержится описание работы с этими функциями. Ниже приведены фрагменты использования этих функций в системе.

```
Wrun = setInterval("doWAIT(0)", 5); window.clearInterval(Wrun);
Lrun = setInterval("vloop(0)", nsleep);
```

```
timerId= setTimeout("func1()",interval1); clearTimeout(timerId);  
timerId= setInterval("func2()",interval2); clearInterval(timerId);
```

На рисунке 4 изображена схема прохождения теста. Тестирование начинается после выбора нужного теста в меню и нажатия кнопки “Старт”. Для каждого теста системы тестирования отображаются свои стимулы. Время ожидания “Интервал 1” и “Интервал 2” могут быть разными. Число отображаемых стимулов в одной итерации теста для каждого теста и блока теста может быть разным. В приведенном рисунке для каждой итерации теста используется только два стимула.

Если испытуемый отвечает на вопрос раньше отведенного интервала времени, тогда прерывается ожидание и вызывается следующая операция.

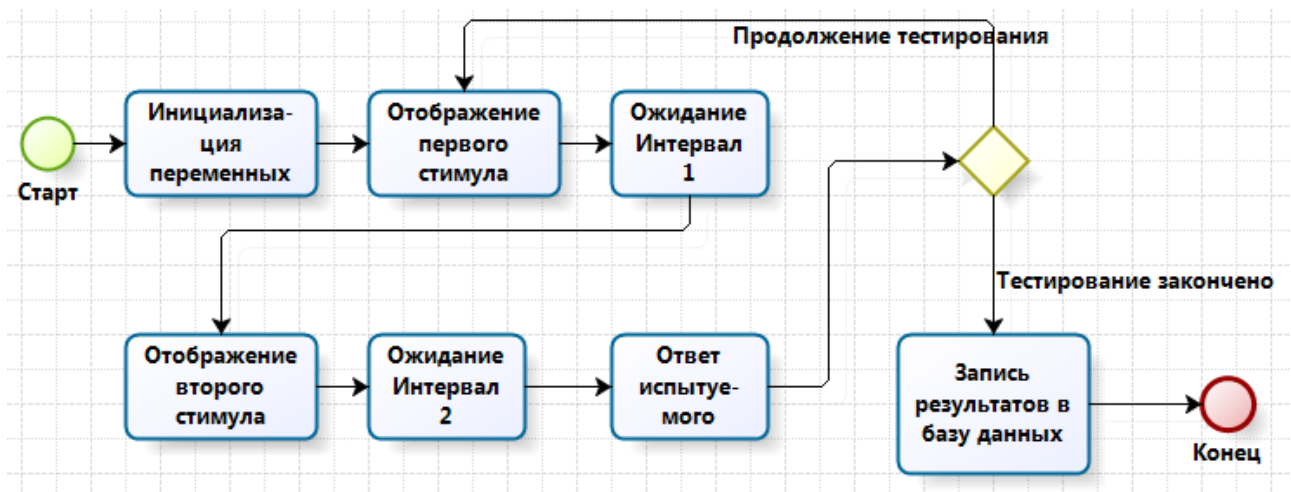


Рис. 4. Упрощенная схема прохождения теста.

### 3.4. Работа со звуком

После завершения некоторых этапов тестирования желательно выдавать звуковой сигнал. Но не все браузеры корректно поддерживают выдачу звукового сигнала. Кроме того (как показал опыт), если на одном компьютере выдача звукового сигнала из JavaScript происходит нормально, то на другом компьютере при использовании такого же браузера звуковой сигнал может отсутствовать. Конечно, можно потратить много времени и выяснить причину происходящего, но часто такого времени нет. Поэтому в системе введён объект “checkbox”, который определяет, следует ли использовать звуковой сигнал. Ниже приведён фрагмент на PHP, который при использовании звука вставляет в файл HTML тег для загрузки звука из файла 'beep.wav'.

```
<?  
if ($zvuk) echo "  
<DIV id=IDbeep style='position:absolute; top:400; left:100; height:100; width:200'  
<embed src='beep.wav' autostart='false' width='0' height='0' id='beep'  
enablejavascript='true'  
</DIV>  
";  
>
```



## ЛИТЕРАТУРА

1. Чельшкова М.Б. Теория и практика конструирования педагогических тестов. Учебное пособие. – М.: Логос, 2002.
2. Нейман Ю.М., Хлебников В.А. Введение в теорию моделирования и параметризации педагогических тестов. – М.: Прометей, 2000.
3. Войтов В. К. Расчет значений сложностей заданий для адаптивного теста интеллекта. – Экспериментальная психология, 2013, том 6, № 2, с. 120–128
4. Войтов В. К. Особенности программной реализации адаптивного теста общего интеллекта Берглинга-Холлинга. – Моделирование и анализ данных, 2013, № 1, с. 110-115.

*Работа поступила 19.02.2014*