

## **КЛАССИФИКАЦИЯ ВИДОВ ПРАКТИЧЕСКИХ РАБОТ И ИХ СООТНОШЕНИЕ С ПРОФЕССИОНАЛЬНЫМИ КОМПЕТЕНЦИЯМИ ПРИ ИЗУЧЕНИИ ДИСЦИПЛИН ПО WEB-РАЗРАБОТКАМ**

**Т.Б. Казиахмедов**

к.п.н., доцент кафедры информатики и методики преподавания информатики,  
e-mail: ktofik@yandex.ru

**Т.В. Мосягина**

аспирант, e-mail: mt.skorp@yandex.ru

Нижевартовский государственный университет

**Аннотация.** Исследования в области подготовки IT-специалиста претерпевают различные подходы разрешения противоречий между ФГОС ВО, профессиональными стандартами и корпоративными стандартами, т. е. функционалом IT-должностей на предприятиях. Настоящее время характеризуется очень быстрой сменой как названия должностей, так и их функционала. Этому способствует бурное развитие инструментария разработок и технологических подходов, связанных с развитием сетевой инфраструктуры и методов оптимизации высоконагруженных сетей. В процессе обучения будущих бакалавров ИВТ необходимо ориентироваться на уровень знаний обучающихся на входе, и обучение базовым и инновационным подходам в разработке web-проектов построить таким образом, чтобы на выходе они знали и новые инструменты и технологии. Процесс обучения web-технологиям и проектированию требует поиска подходящих технологий и методик преподавания, ответов на актуальные и противоречивые вопросы: «Чему и как учить?», «Как учесть требования ФГОС ВО, профессиональных стандартов, корпоративных стандартов, которые порой противоречат друг другу?». В представленной статье рассмотрены некоторые результаты исследований в данном направлении.

**Ключевые слова:** профессиональные стандарты, уровни компетенций, базовые знания и технологии в области Web, инновационные подходы в Web, междисциплинарные знания для web-проектирования, организация взаимодействия между приложениями, авторский программный комплекс обучения web-разработке.

Анализ потребностей специалистов предприятий в области Web показывает, что должности на предприятиях и трудовые функции профессиональных стандартов часто противоречат стандарту ФГОС ВО. Установка, что вузы сами должны адаптировать образовательные программы под IT корпорации с учётом трудовых функций профессиональных стандартов не всегда просто реализуема,

причём не всегда и не по всем инженерным направлениям определены трудовые функции. Развитие инструментария разработки, методов и технологий организации сетей, информационных систем организации и хранения данных способствуют постоянной динамике функционального содержания IT профессий, да и изменению названия самих должностей. Мы часто слышим выражения, что сегодня «программисты» не нужны, причём, понимая под этим человека, владеющего только языком или средой разработки. Если говорить вкратце, то сегодня мы говорим о специалисте (бакалавре, магистре) в области IT, понимая под этим то, что он знает, умеет, применяет на практике:

- методы выделения сущностей, построения логических и физических моделей этих сущностей в задачах автоматизации;
- стандарты и методику формирования технических заданий на проектирование ИС;
- методы моделирования информационных систем (ИС) — ролевою модель, функциональную модель, объектную модель;
- архитектуру ЭВМ и компьютерных сетей, сетевые технологии, инфраструктуру компьютерных сетей, методы организации сетей;
- программное обеспечение ЭВМ и автоматизированных систем;
- технологии реализации ИС (клиент-серверная организация приложений, многозвенная информационная система с распределёнными данными, разные подходы в организации взаимодействия приложений и пользователя с приложениями (интерфейсы взаимодействия с приложениями), СОМ — технология, сервисная организация приложений);
- инструменты и среды разработки ИС и программных комплексов;
- методы тестирования ИС и программных комплексов;
- использование в информационных системах мобильных устройств;
- проблематику интеллектуализации информационных систем;
- робототехнические модули информационных систем и программных комплексов;
- стандарты разработки технической сопроводительной документации ИС и программных комплексов;
- методы обучения персонала;
- психологию коллектива разработчиков;
- и др.

Конечно, на уровне бакалавра не ставится задача подготовки руководителя группы разработки, но, тем не менее, он имеет представление о функциональных обязанностях каждого члена группы разработки. Следовательно, сегодня мы выпускаем не программиста в старом понимании, а специалиста с достаточно широким набором знаний для IT индустрии. На корпоративном уровне мы имеем дело с конкретным программным обеспечением, с конкретной сетевой инфраструктурой, и часто в некоторых фирмах — с собственными программными разработками. Это не создаёт сегодня какую-то проблему, которая ставит под сомнение качество подготовки специалистов для IT индустрии, так как за счёт курсов по выбору мы можем внедрить и изучение корпоративных программных комплексов, и особенностей сетевой инфраструктуры.

Когда мы говорим о Web, то обычно нам приходится ориентироваться на потребности региональных IT предприятий, да и других организаций. Рассмотрим, как обычно выстраиваются должности в области IT индустрии в нашем регионе.

**Тестирование ПО:**

- понимание сферы тестирования программных продуктов (методики, планы, отчёты, чек-листы, тест-кейсы);
- знание основ проектирования реляционных БД;
- умение писать SQL запросы (базовые/начальные знания);
- представления основ проектирования ПО (понимание ЖЦПО);
- понимание и умение работать с технической документацией;
- знание и опыт работы с компьютером на уровне продвинутого пользователя, умение работать в ОС семейства Windows;
- базовые знания объектно-ориентированного программирования;
- опыт в составлении и успешном применении тестовых сценариев;
- знание языков программирования C++ или Java, C#;
- опыт разработки программ;
- опыт проектирования БД, знание языка SQL.

**Аналитик:**

- опыт проектирования и документирования информационных систем, моделирования бизнес-процессов;
- знание стандартов и нотаций визуализации и документирования (UML, IDEF, BPMN);
- знание инструментов моделирования (IBM Rational Rose, ARIS Toolset, BPWin, MS Visio);
- знание методологий разработки и внедрения программного обеспечения;
- опыт работы в области системного анализа (архитектуры, разработки ПО);
- технический английский;
- знание SQL;
- опыт работы с web-сервисами (WSDL, XSD).

**Сопровождение ПО:**

- знания в области Internet-технологий;
- базовые знания телекоммуникации, протоколы TCP/IP, HTTP, SMTP и т. д.;
- знания в области БД, СУБД Oracle, языка PL, SQL, SQL;
- навыки программирования;
- системный подход к решению поставленных задач;
- обучение пользователей основным методам работы с ПП.

**Инженер-программист:**

- опыт работы с базами данных ( MS SQL, InterBase );
- знание 1С (сертификаты);
- Transact-SQL, PL/SQL для баз данных MSSQL и ORACLE;
- ASP/ASP.NET, web-дизайн;
- умение работать с отладчиком для поиска и фиксации ошибок;

- знание языков ООП, таких как C++, C#, Java, JavaScript;
- отличное знание и умение в составлении SQL запросов;
- отличное знание и умение в написании Функций, Триггеров, Хранимых процедур на языке T-SQL;
- понимание взаимодействия приложений по модели клиент-сервер;
- практические навыки работы с Oracle, Delphi, Java;
- обязательное хорошее знание Java 5/6 и популярных фреймворков (Spring Core, Web Services, ORM, сборщики проектов, желательно Maven);
- умение самостоятельно разобраться в чужом коде;
- знание SQL на уровне CRUD запросов;
- способность грамотно составить документацию как по собственной выполненной работе, так и по принимаемым проектам;
- английский на уровне правильного понимания документации;
- знание языков программирования, MS SQL Server 2000/2005, VBA, Autolisp, AutoCAD 2012. Опыт внедрения и сопровождения ПО;
- Java 6, Java EE (JPA, WS);
- Spring Framework;
- Struts 2;
- JavaScript;
- Oracle.

**Веб-программист:**

- Знание Rails-подобных фреймворков: Symfony, Django, Zend;
- понимание принципов MVC;
- опыт работы с реляционными базами данных (Postgres, MySql), noSQL-решениями;
- понимание PHP, умение разбираться в чужом коде при необходимости;
- знание HTML/CSS/JavaScript, AJAX, jQuery;
- опыт работы с БД MySQL;
- опыт создания web-приложений на ASP.NET MVC;
- знание JavaScript/CoffeScript;
- знание SQL;
- опыт работы на C#, ADO (.NET), MS SQL 2005/2008 (database design, T-SQL), NET Framework 2.0, 3.0, 3.5;
- опыт работы в Visual Studio 2010 (знание C# 4.0) или выше;
- опыт web-программирования на ASP.NET и Silverlight 4, JavaScript, jQuery, AJAX, HTML, CSS;
- опыт работы с Web-Services и WCF;
- опыт разработки запросов различной сложности на T-SQL, составление отчётов с использованием Microsoft SQL Reporting Services (SSRS);
- опыт настройки и использования Microsoft Reporting Service.
- опыт коммерческого программирования PHP;
- понимание принципов и практический опыт объектно-ориентированного программирования;
- опыт оптимизации кода и запросов по производительности;
- дополнительное знание прочих языков программирования;

- опыт работы с git;
- понимание методологий программирования: ООП, MVC, reactive;
- уровень владения английским языком — средний и выше.

**Системный администратор:**

- опыт составления спецификаций комплексов серверов одного или нескольких основных производителей (Hewlett-Packard — предпочтительно, IBM, Dell);
- опыт проектирования, пусконаладки или самостоятельного администрирования операционных систем и серверных программных комплексов производства Microsoft Windows Server, Exchange, Forefront, SQL, дополнительный SharePoint или/и VMware vSphere;
- чтение и понимание профильной технической документации на английском языке;
- серверные операционные системы семейства Microsoft Windows Server;
- системы виртуализации VMWare, Citrix;
- принципы подключения комплексов серверов в сети передачи данных на базе IP (технологии teaming, virtual connect etc);
- архитектура и принципы построения вычислительных систем высокой готовности (отказоустойчивость);
- технологии резервного копирования;
- администрирование серверов (1C, MS SQL Server);
- оптимизация производительности БД.

**Разработчик БД (Oracle):**

- знание основ программирования;
- опыт работы с СУБД Oracle;
- опыт работы с SQL;
- опыт разработки пользовательской документации;
- опыт тестирования приложений;
- опыт автоматизированного тестирования;
- понимание жизненного цикла разработки ПО;
- разработка ПМИ и участие в приёмо-сдаточных испытаниях у Заказчика.

Как видим из приведённых должностей, знание Web и интернет-технологий присутствует практически везде. В тексте мы выделили эти компоненты курсивом. А если попытаться все это соотнести с рабочими функциями из профессиональных стандартов, то нам придётся взять эти функции из множества направлений. Это как бы первое противоречие, которое возникло потому, что сегодня нет единого реестра должностей и их функционала [2]. Обычно корпоративные требования не дают нам представления о том, как глубоко необходимо знание того или иного инструмента, технологии. А нам в вузе приходится обратить на это внимание. Таким образом, нам нужно построить систему обучения web-технологиям будущих бакалавров ИВТ с учётом корпоративных требований к знаниям IT-специалистов. Для этого мы предложили 5 уровней формирования соответствующих компетенций и связали их по возможности с рабочими функциями профессионального стандарта. Уровни компетенций нами обозначены следующим образом [1]:

- базовый (пороговый);
- прикладной уровень;
- системный уровень;
- профессиональный уровень или междисциплинарный уровень;
- проблемный уровень.

**Базовый (пороговый) уровень.** Объектная модель web-страницы. HTML. Java Script. Вставка скриптов в HTML-документ. Алгоритмические конструкции языка Java Script, типы данных, функции преобразования типов. Серверные скрипты. Язык и технология PHP. Структура ASP документа. Директивы ASP страницы. Конструирование ASP страниц. Net языки и их использование в качестве языков web-разработки.

**Прикладной уровень:**

Таблицы каскадных стилей. Интерфейс web-страниц. Организация общения с сервером. Методы передачи данных на web-сервер. Элементы управления. Методы создания обработчиков событий. Работа с графикой, таблицами, видео и другими объектами. Обработка баз данных. СУБД MySQL. Система стандартных задач по освоению web-программирования.

**Системный уровень.**

Методы проектирования сайтов. Взаимодействие с офисными технологиями. Реализация основ реляционной алгебры через вложение структур. Взаимодействие 2 и более сайтов. Web-сервер. Основы клиент-серверных приложений на основе web-сервера. Разработка web-приложений средствами визуальных сред программирования. Включение в приложения web-страниц.

**Профессиональный уровень.**

Администрирование серверов Apache, IIS. СУБД MySQL. Язык SQL. Обработка БД через web-страницу, провайдеры баз данных. Технология ASP. Серверные элементы. Управление базами данных. Разработка, тестирование, публикация. Web-сервисы и облачные технологии.

**Проблемный уровень.**

Передача данных из приложения в web-страницы и их публикация на сайте. Разработка серверов и клиентских приложений на основе различных протоколов. Использование Фреймворков. Управление базой данных, компонентами сайта. Тестирование, оптимизация времени загрузки ресурсов.

Предлагаемый нами подход включает 5 видов учебной деятельности «Бакалавра ИВТ»:

- 1) решение частной задачи (метод погружения в знание);
- 2) решение общей задачи через решение частных компонентных проблем (понимание);
- 3) разработка проекта как комплекса, т. е. использование междисциплинарных знаний (применение);
- 4) проектирование профессиональных сайтов с тестированием и внедрением (анализ);
- 5) организация взаимодействия сайтов в сети. Публикация и тестирование сайта. Решение возникающих проблем (синтез).

Причём при погружении в основы инструментов мы опираемся на знания обучающихся по другим дисциплинам. Рассмотрим примерные задания 1 вида.

**Задание:**

Создайте функции, определяющие, являются ли два человека:

- сёстрами;
- братьями;
- дедушкой и внуком (внучкой);
- дядей и племянником (племянницей);
- супругами;
- родственниками.

Разработать программу в виде сайта, функциональной программы, пролог-программы.

### JavaScript.

```
<html>
<head>
</head>
<body>
<script type="text/javascript">
  // - База и стандартные предикаты -----
  var mans = ["Иван", "Артём", "Саша", "Степан", "Кирилл"];
  var woms = ["Наташа", "Маша", "Ира"];
  var mothers = {
    "Ира" : ["Маша", "Наташа", "Степан"],
    "Наташа" : ["Артём", "Иван"]
  };
  var fathers = {
    "Саша" : ["Иван", "Артём"],
    "Кирилл" : ["Ира"]
  }

  function man(value) {
    if(mans.includes(value)) return true;
    return false;
  }

  function wom(value) {
    if(woms.includes(value)) return true;
    return false;
  }

  function isSinglePoint(arr1, arr2) {
    for(var i = 0; i<arr1.length; i++) {
      for(var j = 0; j<arr2.length; j++)
      {
        if(arr1[i] == arr2[j])
        {
          return true;
        }
      }
    }
    return false;
  }
}
```

```
function isFather(ft, ch) {
    if(fathers[ft]) {
        var childs = fathers[ft];
        return childs.inlcudes(ch);
    }
    return false;
}

function isMother(mt, ch) {
    if(mothers[mt]) {
        var childs = mothers[mt];
        return childs.inlcudes(ch);
    }
    return false;
}

function getMotherOf(ch) {
    for (var k in mothers){
        if (mothers.hasOwnProperty(k)) {
            var arr = mothers[k];
            if(arr.includes(ch)) {
                return k;
            }
        }
    }
    return "";
}

function getFatherOf(ch) {
    for (var k in fathers){
        if (fathers.hasOwnProperty(k)) {
            var arr = fathers[k];
            if(arr.includes(ch)) {
                return k;
            }
        }
    }
    return "";
}

// - Дополнительные предикаты-----

function isSister(v1,v2) {
    return wom(v1) && wom(v2)
        && (getMotherOf(v1) == getMotherOf(v2));
}

function isBrother(v1,v2) {
    return man(v1) && man(v2)
        && (getMotherOf(v1) == getMotherOf(v2));
}

function isMarried(v1, v2) {
```



```
        return man(v1) && wom(v2)
           && isSinglePoint(fathers[v1],mothers[v2]);
    }

function isRodstv(v1, v2) {
    if(isSister(v1,v2))
        return true;
    else
    {
        if(isBrother(v1,v2))
            return true;
        else {
            var m1 = getMotherOf(v1);
            var m2 = getMotherOf(v2);

            if(m1 != "" && m2!= "") {
                return m1==m2;
            }
        }
    }
    return false;
}

function isDyadya(v1, v2) {
    z = getMotherOf(v2);
    if(isRodstv(v1,z))
    {
        return true;
    }
    return false;
}

function isDedushka(v1, v2) {
    var z = fathers[v1];
    var z2 = mothers[getMotherOf(v2)];
    if(!z)return false;
    if(!z2)return false;

    if(z.includes(getMotherOf(v2)))
        return true;

    return false;
}

// - Функции HTML страницы -----
function printDB()
{
    //Написать базу данных
    var bd = "Ира мать Наташи, Маши, Степана<br>";
    bd+="Наташа мать Артёма и Ивана<br>";
    bd+="Саша отец Ивана и Артёма<br>";
    bd+="Кирилл отец Иры<br>";
    var msg = '<font color="green"> ${bd} </font>`;
    document.getElementById("div_0").innerHTML = msg;
}
```

```
}
function run() {
    var v1 = document.myForm.var1.value;
    var v2 = document.myForm.var2.value;
    var result = isSister(v1,v2);
    var answer = (result === true) ? "Да" : "Нет";
    var msg = '<font color="blue">Ответ: ${answer}</font>';
    document.getElementById("div_1").innerHTML = msg;
}

function run2() {
    var v1 = document.myForm.var1.value;
    var v2 = document.myForm.var2.value;
    var result = isBrother(v1,v2);
    var answer = (result === true) ? "Да" : "Нет";
    var msg = '<font color="blue">Ответ: ${answer}</font>';
    document.getElementById("div_1").innerHTML = msg;
}

function run3() {
    var v1 = document.myForm.var1.value;
    var v2 = document.myForm.var2.value;
    var result = isDedushka(v1,v2);
    var answer = (result === true) ? "Да" : "Нет";
    var msg = '<font color="blue">Ответ: ${answer}</font>';
    document.getElementById("div_1").innerHTML = msg;
}

function run4() {
    var v1 = document.myForm.var1.value;
    var v2 = document.myForm.var2.value;
    var result = isDyadya(v1,v2);
    var answer = (result === true) ? "Да" : "Нет";
    var msg = '<font color="blue">Ответ: ${answer}</font>';
    document.getElementById("div_1").innerHTML = msg;
}

function run5() {
    var v1 = document.myForm.var1.value;
    var v2 = document.myForm.var2.value;
    var result = isMarried(v1,v2);
    var answer = (result === true) ? "Да" : "Нет";
    var msg = '<font color="blue">Ответ: ${answer}</font>';
    document.getElementById("div_1").innerHTML = msg;
}

function run6() {
    var v1 = document.myForm.var1.value;
    var v2 = document.myForm.var2.value;
    var result = isRodstv(v1,v2);
    var answer = (result === true) ? "Да" : "Нет";
    var msg = '<font color="blue">Ответ: ${answer}</font>';
    document.getElementById("div_1").innerHTML = msg;
}
```

```

</script>

<form name = "myForm" onload="printDB()">
  <h1>Лабораторная 1</h1>
  <h2>Создайте функции, определяющие являются ли два человека:</h2>
  <br>
  <b>Человек 1:</b>
  <input type="text" name="var1" size=6 maxlength=8>
  <br><br>
  <b>Человек 2: </b>
  <input type="text" name="var2" size=6 maxlength=8>
  <br><br>
  <input type = "button"
    name = "button1"
    value = "Сёстрами"
    onclick="run()"><br><br>
  <input type = "button"
    name = "button1"
    value = "Братьями"
    onclick="run2()"><br><br>
  <input type = "button"
    name = "button1"
    value = "Дедушкой и внуком(внучкой)"
    onclick="run3()"><br><br>
  <input type = "button"
    name = "button1"
    value = "Дядей и внуком(внучкой)"
    onclick="run4()"><br><br>
  <input type = "button"
    name = "button1"
    value = "Супругами"
    onclick="run5()"><br><br>
  <input type = "button"
    name = "button1"
    value = "Родственниками"
    onclick="run6()"><br><br>
  База данных:
  <div id="div_0"><script>printDB();</script></div>
  <br><br>
  <div id="div_1"></div>
  <br><br>
</form>
</body>
</html>

```

Такие задания студенты имели при изучении основ логического программирования (Пролог) и функционального программирования (Лисп, Питон). Это даёт представление о том, что на самом деле возможности Java Script уникальны, что это вполне серьёзный высокоуровневый язык, который позволяет писать программы в логическом и функциональном стилях. Кроме того, такой подход даёт очень важное понимание по расположению и вызовам скриптов. Такой набор заданий с разными видами деятельности нами реализованы по HTML, PHP, MySQL, ASP.Net. Все материалы, среда выполнения сфокусиро-

**Создайте функции, определяющие являются ли два человека:**

Человек 1:

Человек 2:

База данных:

Рис. 1. Результат выполнения программы в браузере

ваны в виде одного программного комплекса. Приведём некоторые скриншоты программного комплекса.

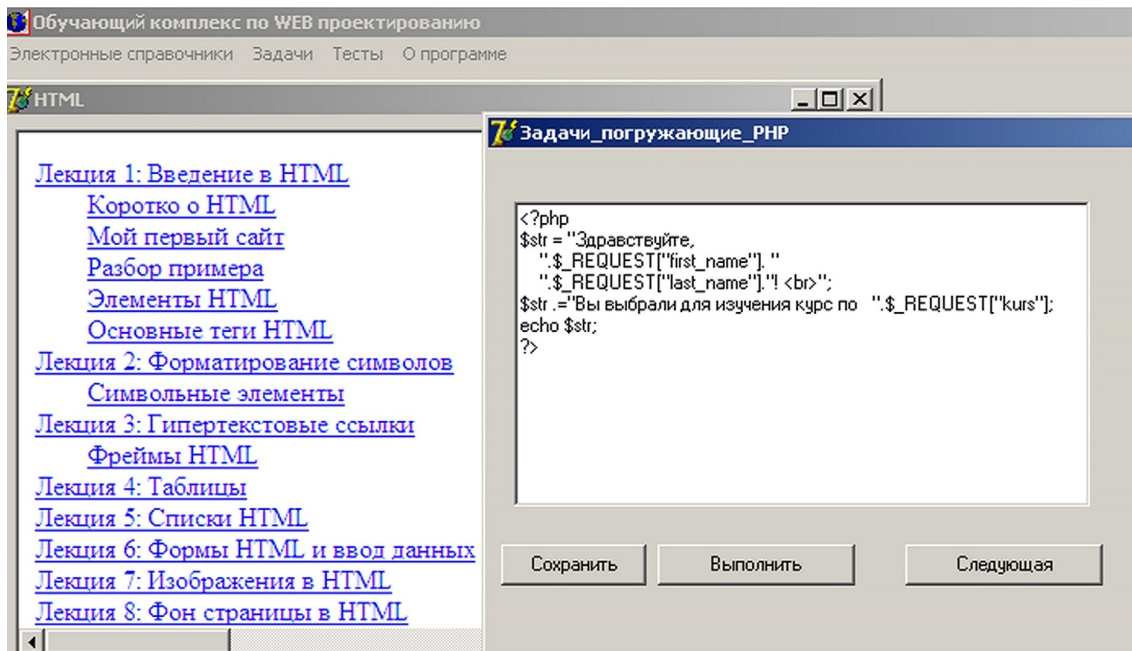


Рис. 2. Скриншот программного комплекса «Обучающий комплекс по web-проектированию»

Данный программный комплекс содержит справочную информацию, методические приёмы и подходы к уровневому обучению бакалавров, систему тестирования, основанную на рабочих функциях профессионального стандарта и

профессиональных компетенциях ФГОС ВО, интеллектуальный модуль, определяющий уровни профессиональных компетенций и предлагающий обучающемуся ту или иную сферу деятельности, а также предлагает углубление по необходимым разделам дисциплин по WEB. Сам программный комплекс адаптируется к любому направлению обучения. Он состоит из 4 модулей:

1. Рабочие функции и профессиональные компетенции.
2. Формирование учебного материала.
3. Система контроля и самоконтроля.
4. Система логического вывода.

Первый модуль позволяет адаптировать систему к любому направлению подготовки, по которому имеются рабочие функции профессионального стандарта. Модуль 2 позволяет сформировать учебно-методические материалы, соотнесённые с разными уровнями профессиональных компетенций (лекции, презентации, видео, задания, практикумы и т.д.). Самым трудно адаптируемым является система логического вывода. Это результат экспертного анализа и экспертных оценок. Разработанный нами программный комплекс апробирован в системе подготовки бакалавров ИВТ, сейчас идёт адаптация к другим инженерным и педагогическим направлениям.

## ЛИТЕРАТУРА

1. Казиахмедов Т.Б. Региональный и национальный аспекты обучения информатике: методология, методика, информационные ресурсы: Монография. Нижневартовск : Изд-во НГГУ, 2010. 143 с.
2. Казиахмедов Т.Б. Опережающее обучение в области индустрии информационных технологий в условиях развивающейся экономики и перманентных реформ высшего образования: Педагогическая информатика. Издательство: Московский государственный гуманитарный университет им. М.А. Шолохова (Москва), ISSN: 2070-9013, 2014. С. 62–72.

**CLASSIFICATION OF JOB TYPES AND THEIR CORELATION  
WITH PROFESSIONAL COMPETENCES WHEN STUDUING DISCIPLINES  
ON THE BASIS OF WEB PROGRAMMES**

**T.B. Kaziakhmedov**

Ph.D. (Ped.), Associate Professor, e-mail: ktofik@yandex.ru

**T.V. Mosyagina**

Postgraduate Student, e-mail: mt.skorp@yandex.ru

Nizhnevartovsk State University

**Abstract.** Research in the field of IT specialists training uses various approaches to resolving conflicts between Federal State Standards of Higher Education (FGOS VO), professional and corporate standards, i.e. IT specialists' functions at enterprises. The present moment is characterized by very rapid changes both in job titles and the personnel functions. This happens because of rapid development of software tools and technological approaches connected with the development of network infrastructure and methods of high-load networks optimization. When training Bachelors of IT, it is necessary to take into consideration the students' initial level of knowledge. Teaching of basic and innovative principles in Web project development should be so planned that the students know both new Web design tools and technologies when graduating. It is urgent to find suitable techniques and methods as well as answers to the topical and controversial questions: "What and how should we teach?", "How can we take into account the requirements of educational, professional and corporate standards which sometimes contradict each other?" This paper presents some results of the authors' research in this area.

**Keywords:** professional standards, levels of competences, basic knowledge and technologies in the field of Web, innovative approaches in Web, multidisciplinary knowledge for Web design, organization of applications interaction, proprietary technology of Web design training.