**Государственная академия наук**

**Российская академия образования**

**Институт информатизации образования**

**В.В. Михаленок**

## Методические подходы к обучению специалистов в области информатики созданию и использованию управляемых сервисно-ориентированных приложений в рамках курса «Web-ориентированная платформа .NET»

**Москва**

**2010**

## Михаленок В.В. Методические подходы к обучению специалистов в области информатики созданию и использованию управляемых сервисно-ориентированных приложений в рамках курса «Web-ориентированная платформа .NET». - 2-е изд. – М.: ИИО РАО, 2010.

Учебный и практический материал данного пособия структурирован таким образом, что сначала формули­руется тема и цели конкретной практической работы, затем изла­гается теоретический материал, необходимый для выполнения за­даний к данной работе, далее предлагаются практические задания с технологией их выполнения. Цель всего практикума - переход от репродуктивного уровня выполнения студентами учебных за­даний к творческому (при создании собственных приложений).

© ИИО РАО, 2010.

**Содержание**

1. Методические подходы к обучению специалистов в области информатики созданию и использованию управляемых сервисно-ориентированных приложений. …………………………………………4
2. Требования к уровням обученности специалистов в области информатики созданию и использованию управляемых сервисно-ориентированных приложений в рамках курса «Web-ориентированная платформа .NET»………………………………………………………...19
3. Библиография …………………………………………………………….29

## Методические подходы к обучению специалистов в области информатики созданию и использованию управляемых сервисно-ориентированных приложений.

Для успешного изучения учебного материала преподавателю необходимо знать, как происходит процесс усвоения знаний, какими особенностями он характеризуется на каждом этапе обучения.

Наиболее полно и конструктивно закономерности процесса усвоения представлены в деятельностной теории учения, известной под названием теории поэтапного формирования умственных действий, которая заложена в работе П.Я. Гальперина [3]. В исследовании Н.Ф. Талызиной [7] выделены ориентировочная, исполнительская, контрольная и корректировочная части действия. Ориентировочная основа действия – эта та система условий, на которую реально опирается человек при выполнении действия. Ориентировочная часть действия направлена:

* на правильное и рациональное построение исполнительской части;
* на выборе одного из возможных исполнений.

П.Я. Гальперин [3] выделил три типа ориентировочной основы действия. Ориентировочную основу первого типа составляют только образцы – действия и его продукты. Никаких указаний, как правильно выполнить это действие не дается. Ориентировочная основа второго типа содержит не только образцы действия и его продукта, но и все указания на то, как правильно выполнить действие с новым материалом. Ученик при этом приобретает умение анализировать материал с точки зрения предстоящего действия, и это ведет к тому, что последнее обнаруживает заметную устойчивость к изменению условий и переносится на новые задания.

Ориентировочная основа третьего действия отличается тем, что здесь на первое место выступает планомерное обучение такому анализу новых заданий, который позволяет выделить опорные точки, условия правильного выполнения заданий. Затем по этим указаниям происходит формирование действия, отвечающего данному заданию. Этот тип ориентировки направлен на познание сущности, является основой формирования теоретического мышления. Однако, какой бы ни была ориентировочная основа действия, она все-таки остается лишь системой указаний на то, как выполнить новое действие, и не является самим действием.

Контроль – это неотъемлемая часть обучения. В зависимости от функций, которые выполняет контроль в учебном процессе, выделяют предварительный, текущий и итоговый контроль. Назначение предварительного контроля состоит в установлении исходного состояния познавательной деятельности. В курсе «Web-ориентированная платформа .NET» необходимо выделить знания, которые относятся как непосредственно к программированию (использование объектно-ориентированной техники программирования с использованием языков высокого уровня, умение составлять алгоритмы), так и к созданию Web-содержания (сайты, порталы) и основам баз данных (проектирование и реализация баз данных, составления запросов к базам данных).

Как известно, важнейшей функцией текущего контроля является функция обратной связи. Обратная связь позволяет преподавателю получать сведения о ходе усвоения у каждого учащегося. Она составляет одно их важнейших условий успешного протекания процесса усвоения. Обратная связь должна нести следующую информацию:

* выполняет ли обучаемый то действие, которое намечено;
* правильно ли обучаемый выполняет намеченное действие;
* соответствует ли форма действия данному этапу усвоения;
* формируется ли действие с должной мерой обобщения, освоения (автоматизированности, быстроты выполнения и др.) и т.д.

Результаты исследований Н.Ф. Талызиной [7] показывают, что частота контроля зависит от этапа усвоения и должна меняться внутри отдельных этапов от систематического внешнего контроля (обучающего или заменяющего его технического устройства) до самоконтроля обучаемого на завершающих этапах становления деятельности.

Итоговый контроль используется для оценки достигнутых результатов обучения в конце работы над темой или курсом. Функция итогового контроля состоит в том, каком уровне учащийся усвоил изучаемый материал. Так, В.П. Беспалько [2], основываясь на содержании деятельности, в которой должны использоваться усваиваемые знания, выделяет четыре уровня усвоенности знаний: уровень узнавания, уровень воспроизведения, уровень применения знаний в привычных условиях и уровень применения в новых условиях (творческое применение знаний).

При определении заданий итогового контроля Н.Ф. Талызина [7], опираясь не теорию деятельностного подхода к процессу обучения, руководствоваться следующими этапами:

* при определении содержания нового материала необходимо определить задачи, при решении которых этот материал будет использоваться учащимися;
* с помощью задач установить, в каких познавательных действиях учащиеся должны уметь использовать данные знания;
* указать, какими качествами должны обладать выделенные познавательные действия и входящие в них знания.

На основе анализа и синтеза знаний и умений, приобретаемых во время обучения, были выделены два этапа обучения специалистов: «Основные технологии платформы .NET» и «Метод проектов».

Рассмотрим первый этап – «Основные технологии платформы .NET».

Целью первого этапа курса «Web-ориентированная платформа .NET» является:

1. познакомить обучаемого с основными технологиями, предоставляемыми платформой .NET;
2. сформировать у обучаемого навыки разработки приложений на основе иерархии объектов.

Для реализации этих целей необходимо выделить основные технологии. Из всего многообразия технологий, рассмотрим следующие:

* программная среда .NET Framework и ее основные компоненты (CLR, FCL, сборка мусора);
* технология создания настольных Windows-приложений WinForms;
* технология создания Web-приложений ASP .NET;
* технология доступа к данным ADO .NET;
* технология XML Web-Services;

Как показала практика, изучение технологий целесообразно выполнять в соответствии со следующей схемой:

* + - проблемы, решаемые с использованием данной технологии;
    - классы, составляющие данную технологию, их иерархия и способы использования;
    - возможности использования данной технологии.

В содержание обучения программированию с использованием технологий платформы .NET включены следующие темы:

1. Основы объектно-ориентированного программирования. Использование языка программирования C#.

2. Среда разработки Visual Studio.NET.

3. Программирование с использованием возможностей среды .NET FRAMEWORK.

4. Создание безопасных приложений.

5. Использование технологий баз данных при создании приложений. Технология ADO .NET.

6. Создание Windows-приложений. Технология WinForms.

7. Создание Web-приложений. Технология ASP .NET.

8. Создание приложений с использованием технологий XML Web-services.

На основании дидактических принципов обучения в вузе, требований государственных образовательных стандартов высшего профессионального образования были выделены знания, умения и навыки, которыми должен овладеть будущий учитель.

*Знания*. Структура программы на C#. Встроенные типы. Выражения. Условные операторы и операторы цикла. Введение в классы. Ссылочные и структурные типы. Пространства имен. Модификаторы доступа. Конструкторы класса. Статические методы класса. Наследование. Исключения и техника обработки исключений. Пользовательские исключения. Делегаты и события. Свойства и индексаторы. Атрибуты. Пользовательские атрибуты. Знакомство с управляемой средой выполнения. Архитектура .NET. Common Language Runtime. Управляемый и неуправляемый код. Типы угроз и уязвимостей программного обеспечения. Средства безопасности, предоставляемые средой .NET Framework. Безопасный доступ кода (Code Access Security). Безопасность, основанная на ролях (Role-based Security). Криптография. Управляемые поставщики данных. Присоединенная среда для работы с данными. Знакомство с классом DataSet. Связь с данными в компонентах.

*Умения*. Объявление класса и создание объекта. Создание и использование методов класса. Перегрузка методов. Переопределение методов. Объявление интерфейсов. Использование исключений. Массивы. Создание и использование делегатов. Создание и использование событий. Работа с типами. Основы управления памятью в среде .NET Framework. Создание сборок. Использование сборок. Введение в базовые классы .NET Framework. Развертывание и внедрение .NET приложений. Контроль версий. Перенос существующих приложений в .NET. Разработка безопасных приложений. Извлечение и обработка данных. Работа с данными в отсоединенной среде.

*Навыки*. Проектирование интерфейса приложения с использованием среды разработки Visual Studio .NET. Использование обработчиков событий. Отладка приложения.

Для формирования опыта творческой деятельности, а также сознательного усвоения знаний и способов действий наиболее целесообразно использовать методы продуктивного характера, такие как метод проблемного изучения, эвристический метод и исследовательский метод. Если в учебной работе используются такие методы, то познание предстает перед обучаемым не как процесс накопления информации, а как процесс постановки и поиска решения различных научно-познавательных и практических проблем.

Проблемное изложение позволяет проследить логику этого, подчас противоречивого, поиска. Эвристический метод дает возможность поэлементного усвоения опыта творческой деятельности. Исследовательский метод заключается в организации поисковой творческой деятельности обучаемых по решению проблем и проблемных задач.

Освоение индивидом опыта творческой деятельности тесно связано с его развитием. Развитие личности обучаемого, его способностей, творческого потенциала – тот приоритет, который признается сегодня педагогической наукой.

Применение задачного подхода к организации процесса обучения предполагает работу с соответствующей задачей на каждом этапе обучения. Значит, система задач должна быть подобрана таким образом, чтобы обучаемый, последовательно переходя от решения одной задачи к другой, мог достичь целей, поставленных в отношении развития его способностей и формирования определенных знаний и умений. Система задач является организующим стержнем и в полной мере исполняет функцию управляющего воздействия в процессе обучения. Можно предложить следующий порядок формирования системы задач для изучения некоторой конкретной темы. Решение задачи должно быть направлено на формирование нескольких основных способов действий. Формулировка ее проста и явно указывает на то, какие именно умения предполагается сформировать. Одна или несколько следующих задач служат закреплению и упрочению сформированных основных способов действий.

Решение каждой последующей задачи предполагает усвоение учеником еще одного нового способа действий. При этом в каждой из задач закрепляются способы действий, усвоенные ранее. Формулировки становятся более общими, в них отсутствуют явные указания на необходимость применить тот или иной, уже освоенный учеником, способ действий – об этом еще предстоит догадаться. Такая работа способствует проявлению творческой активности и сознательности усвоения материала. В то же время задачи приобретают большую практическую направленность, что позволяет ученику применять уже усвоенные способы действий в реальных ситуациях и усиливает мотивацию. Также каждая задача должна способствовать приобретению обучаемым нового знания, которое он либо получает непосредственно в ходе ее решения, либо весь процесс решения мотивирует необходимость получения такого знания либо от учителя либо из других источников.

Остановимся на втором этапе обучения, предполагающем выполнение учебных задач, и названным методом проектов

Обучение программированию осуществляется посредством решения задач. Понятие учебной задачи, структура задач, их место в деятельности субъекта, задачный подход к организации учебного процесса рассматриваются в трудах Г.А. Балла [1], И.Я. Лернера [4], Д.Б. Эльконина [6] и других. Задачный подход предполагает, в частности, проектирование системы учебных задач, удовлетворяющих заранее намеченным требованиям, в разработке которой целесообразно использовать основные положения теории учебных задач [1].

При таком подходе деятельность обучаемых может быть представлена как система процессов решения разнообразных задач. При этом понятие «задача» трактуется достаточно широко, и процесс обучения может быть рассмотрен как процесс разрешения противоречий «между выдвигаемым ходом обучения, познавательными и практическими задачами, начальным уровнем знаний, умений и умственного развития учащихся» [1].

Чтобы перечисленные функции учебной задачи могли быть реализованы, задача конструируется учителем специально, являясь, по выражению И.Я. Лернера [4], «искусственной педагогической конструкцией», и в определенный момент включается в учебный процесс.

Учитель, разрабатывая задачу, предусматривает:

* способы действий, которые требуется сформировать в процессе ее решения;
* новые знания, которые ученик приобретает во время поиска и анализа решения задачи;
* творческие процедуры, которые необходимы для поиска решения.

В своей работе над системой заданий мы руководствовались следующими положениями:

* задания должны быть построены таким образом, чтобы их выполнение требовало применения знаний и умений по информатике;
* система заданий должна отражать содержание педагогической деятельности;

условия заданий должны предусматривать возможность их выполнения с использованием визуальной технологии.

Рассмотрим второй этап – «Метод проектов».

В настоящее время наиболее распространенный способ закрепления лекционного материала заключается в том, что на практических и лабораторных занятиях студенты разбирают и решают тщательно подобранные задачи, соответствующие изучаемой теме. Такой подход помогает усваивать материал небольшими порциями, но не решает проблем, возникающих при разработке больших программ, так как подборка задач не связана обычно какой либо стержневой темой. Решением проблемы может стать использование метода проектов. Методика использования метода проектов при обучении информатике в педагогическом вузе и школе рассматривалась в работах Э.Т. Селивановой, Н.Ю. Пахомовой и др.

Н.Ю. Пахомова [8], показала что, метод проектов осмысленный в ретроспективе его использования в педагогической практике, при его современном осмыслении и принятии как компонента системы образования, имеет богатые дидактические возможности далеко не полностью исследованные и используемые, как для внутрипредметного, так и межпредметного обучения. Поиски средств активизации познавательной деятельности учащихся, развития самостоятельности, обучения приемам мышления и деятельности приводят к пересмотру дидактических возможностей методов проектов, форм его реализаций, поиску методики использования учебных проектов в преподавании различных школьных предметов и в первую очередь предмета информатика.

При обучении информатике метод проектов позволяет использовать все его воспитательные и дидактические возможности, но получает свои акценты значимости. «На уроках при изучении отдельной темы или крупного тематического блока содержания раздела «Элементы программирования» метод проектов может быть использован как один из методов проблемного изучения, активизирующий и углубляющий познание, как метод позволяющий обучать самостоятельному мышлению и деятельности в процессе обучения предметной деятельности, групповому взаимодействию (что очень важно для информатики - предмета с ярко выраженной практической направленностью)» [8]. В рамках проектной деятельности обучаемый может реализовать уже накопленные знания, практически применить их, проявить самостоятельность творчества, у него проявляется достаточно свободы для самовыражения и самореализации. Работа над проектом, как правило, носит поисковый характер, позволяет научить системному подходу в решении проблемы проекта (для информатики системный подход в решении задач является учебным умением, которое должно вырабатываться в процессе обучения).

Перечислим принципы обучения с помощью учебных проектов:

* взаимодействие учебно-воспитательных целей обучения и цели проекта при обеспечении мотивации работы учащихся;
* создание условий бесконфликтной педагогики для воспитания самокритичности, обучения самоанализу и рефлексии;
* групповая работа учащихся и организационные формы проведения обучения методом проектов в условиях классно-урочной системы учебно-воспитательного процесса в школе.

Второй этап изучения технологий платформы .NET назван «Методом проектов», так как учащиеся уже знакомы с основными технологиями платформы .NET. На этом этапе изучения возможностей среды учащиеся, используя полученные на первом этапе знания, используют их для реализации технологий, реализованных в операционной системе Windows.

Перечислим перечень знаний и умений, которыми должен овладеть будущий специалист в области информатики.

*Знания*. Этапы разработки компонент: выбор класса предка; создание модуля компоненты; добавление в новую компоненту свойств, методов и событий; тестирование; регистрация компоненты в среде Visual Studio .NET; создание системы помощи для приложений. Сборки с общим именем. Создание и внедрение сборок с общим именем. Создание и использование библиотек. Исключительные ситуации. Многозадачность и многопоточность. Управление памятью в среде .NET Framework и сборщик мусора: виртуальная память, отображаемые в памяти файлы, кучи. Пространство имен System.Threading: принципы работы. Критические секции, мьютексы, семафоры как методы синхронизации потоков. Приоритет потоков.

*Умения:* использовать функции библиотеки классов .NET Framework при разработке приложений; создавать многопоточные приложения; разрабатывать библиотеки динамической компоновки и использовать их в приложениях; создавать компоненту и размещать ее в палитре компонент. Использовать новые компоненты при разработке приложений.

Приведем примерный перечень тем учебных проектов.

1. Технология Drag&Drop. Буфер обмена.

Операционная система Windows широко использует специальный приём связывания программ с данными, который называется Drag&Drop (перетащи и отпусти). Такой приём в Проводнике Windows используется для копирования или перемещения файлов, а также для запуска обрабатывающей программы. Если, например, файл с расширением DOC «перетащить» на пиктограмму WinWord, автоматически запустится текстовый редактор Word и в окне появится текст из этого файла.

Для демонстрации использования технологии Drag&Drop и буфера обмена можно воспользоваться разработанным ранее текстовым редактором.

Для реализации собственного приложения можно предложить учащимся разработать задачу, которую они хотели бы решить. Прием придумывания задач активизирует творческую деятельность обучаемых, подготавливает их к преподавательской деятельности, умению подобрать задачу по теме занятия, либо самому разработать ее. При выполнении этого задания обучаемые воспользовались изучением занимательной литературы, придумыванием собственной задачи, формулировкой задачи, программную реализацию которой уже видели. Как показала практика, после выполнения этого при изучении новых тем обучаемые стараются сами придумать задания для решения.

2. Использование технологий COM в .NET приложениях.

Технология COM – это компонентная технология и интерфейс для взаимодействия объектов; способ передачи и разделения информации между программами.

При изучении этой темы осуществим передачу данных из приложения, реализованного в среде .NET Framework, в документ пакета MS Office. Учащиеся познакомятся с языком программирования VBA (Visual Basic for Application), а также с существующими в пакете MS Office объектами. В качестве примеров задач по этой теме можно привести следующие:

* разработать приложение для графического отображения сведений об объемах продаж в пяти регионах за последние пять лет с использованием объектов Excel (заданные в файле данные перенесите на лист Excel, оформить в виде таблицы и построить диаграмму в Excel);
* составить приложение для проверки орфографии в написанном средствами MS Word диктанте.

3. Разработка приложения, предназначенного для поиска файлов.

Рассмотрим сценарий работы приложения. Пользователь выбирает путь, где следует проводить поиск, указывает маску файла, чтобы уточнить тип искомых файлов. Кроме того, в соответствующую строку редактирования пользователь вводит лексему поиска. В форме имеются специальные флажки опций, с помощью которых можно указать специальные условия поиска (поиск в поддиректориях, чувствительность к регистру, и др.). По щелчку на кнопке «Поиск» создается поток поиска и в объекте потомка класса TThread передается информация, необходимая для поиска. Когда поток в определенном файле обнаруживает искомый, в окно списка добавляется соответствующая информация. По двойному щелчку на имени файла в окне списка, соответствующий файл открывается с помощью текстового редактора или другой связанной с ним программы.

При разработке приложения учащимся необходимо самостоятельно разобраться с функциями WinAPI, реализующими работу с файловой системой.

4. Создание оригинальных компонент в среде Visual Studio .NET.

При разработке приложений студенты знакомились с принципами объектно-ориентированного программирования языка программирования C#. При разработке собственных компонент обучаемый показывает знания:

* принципов объектно-ориентированного программирования, умение их использовать при разработке программ;
* иерархии объектов среды .NET Framework;
* процедур и функций Windows API.

Для самостоятельного выполнения можно предложить следующие упражнения:

* разработать компонент - трехмерная метка;
* разработка компонента – окно редактирование с выравниванием по правому краю;
* разработка компонент для построения графиков функций.

При выполнении упражнений учащиеся используют лекционный материал и файл помощи среды .NET Framework.

Итак, второй уровень усвоения понятий «Метод проектов» преследует цель – использовать знания, полученные на первом этапе, для самостоятельного изучения технологий, изучаемых на лекции.

Заключительный этап обучения характеризуется тем, что обучаемые уже имеют навыки работы с основными технологиями платформы .NET и обладают необходимыми знаниями для реализации курсового проектирования.

Методика курсового проектирования предусматривает решение обучаемыми задачи, формулируемой в какой-либо проблемной области, требующей формализации и последующего его решения с помощью ЭВМ. Такая задача, как правило, требует значительного времени для решения, системного подхода при разработке, имеет большой объем программирования. Она требует знания технологии решения любой задачи с помощью ЭВМ: умение увидеть проблемы, решаемые задачей, в ряду проблем рассматриваемой области; умение перейти от проблемной постановки задачи к математической с последующей алгоритмизацией и программированием ее решения; умение проанализировать полученные результаты с точки зрения обозначенных проблем. Это, так называемые, большие задачи.

Для подготовки углубленного изучения программирования необходимо практическое решение большой задачи с выполнением всех этапов изучаемой технологии. В процессе работы над большой задачей можно изучить и практически использовать приемы программирования больших задач, реализовать современный стиль программирования.

В качестве курсового проекта решаются реальные задачи, предполагающие последующее внедрение. В процессе работы над курсовым проектом происходит:

* вовлечение специалистов в области информатики в реальную предметную область, породившую задачу проекта;
* расширение знаний в процессе выполнения проекта;
* достижение результата обеспечивает качественно новое состояние компетентности специалиста в предметной области, поскольку проектирование направлено на совершенствование подготовки по программированию;
* развитие навыков самостоятельной работы в процессе выполнения проекта;
* приобщение к технологиям реальной деятельности в области программирования.

В качестве курсовых проектов можно предложить задачи различной тематики. Приведем некоторые из них:

1. Разработка внутреннего портала группы, позволяющего использовать основные технологии .NET.

Обучаемым необходимо разработать комплекс программных блоков по теме исследования (программа-сервер, программа-клиент, информационные Web-сервисы, безопасная передача данных).

Результат выполнения курсового проекта – портал группы.

Работа над выполнением проекта требует от обучаемых знаний в области теоретических основ информатики, сетевого программирования. В перспективе программа может быть использована при проведении уроков по программированию.

2. Разработка сервера баз данных.

Использование баз данных – это самая распространенная область применения компьютеров. С распространением компьютерных сетей стала актуальной проблема использования информации из баз данных с удаленных компьютеров, возникла необходимость разработки клиент-серверных баз данных.

Этапы выполнения курсового проекта:

* знакомство с принципами работы существующих серверов баз данных;
* разработка системы безопасности разрабатываемого сервера;
* реализация сервера в среде .NET Framework.

При разработке проекта обучаемым необходимо изучить возможности среды программирования Visual Studio .NET для работы с базами данных, исследовать средства создания сетевых приложений. Реализация этого проекта обеспечивает более глубокое знакомство с принципами организации баз данных, что обеспечивает взаимосвязь с курсом «Информационные системы».

После реализации курсовых проектов проводится их защита. Одной из задач, стоящих перед курсовым проектом – научить будущего специалиста в области информатики описывать инструкции по программе.

Итак, курсовой проект – заключительный уровень усвоения понятий по курсу «Web-ориентированная платформа .NET». Реализация курсового проекта – самостоятельная творческая научная работа студента, которая требует от него не только знания среды программирования, но и навыков работы со специальной литературой. Работа над курсовым проектом развивает самостоятельность обучаемых, демонстрирует умения и навыки, которые он приобрел при усвоении основных понятий среды на первом и втором этапах выполнения курсового проекта.

## 

**2.Требования к уровням обученности специалистов в области информатики использованию технологий платформы .NET при создании управляемых сервисно-ориентированных приложений**

Обучение специалистов в области информатики курсу «Web-ориентированная платформа .NET» должно обеспечивать необходимый каждому специалисту начальный уровень овладения технологиями, предоставляемыми платформой .NET.

Для оценки результатов обучения специалистов в области информатики предлагаемому курсу, на основе выделенных в первой главе основных направлений подготовки специалистов в области информатики, определим уровни обученности использования технологий платформы .NET при создании управляемых сервисно-ориентированных приложений.

В качестве основы для выделения уровней обученности выберем степень самостоятельности и осознанности действий [2] при создании управляемых сервисно-ориентированных приложений на базе технологий платформы .NET. Вслед за А.Е. Шухманом [5], адаптировав предложенную В.П. Беспалько систему уровней освоения действий, выделим четыре уровня овладения способами информационной деятельности по созданию управляемых сервисно-ориентированных приложений – репродуктивный, адаптивный, эвристический и творческий.

На репродуктивном уровне овладения способами информационной деятельности специалисты в области информатики обладают отрывочными знаниями в области использования различных технологий, предоставляемых платформой .NET для создания управляемых сервисно-ориентированных приложений, а также применяют данные технологии не вполне осознанно – по заданному алгоритму или копируют действия других. У них отсутствуют или слабо выражены мотивы использования технологий современных Web-ориентированных платформ.

На адаптивном уровне специалисты-информатики осознанно и самостоятельно применяют полученные знания, усвоенные способы (алгоритмы) использования технологий Web-ориентированных платформ для типичных ситуаций; проявляют частичную самостоятельность в процессе решения поставленной задачи; у них присутствует направленность на использование Web-ориентированных платформ для решения поставленных задач;

На эвристическом уровне обучаемые имеют глубокие и прочные знания и умения; могут самостоятельно и эффективно использовать технологии, предоставляемые Web-ориентированными платформами для разработки приложений, правильно выбрав усвоенные методы их применения и адаптировав их к конкретной задаче; у них присутствует направленность на использование Web-ориентированных платформ в профессиональной и учебной деятельности, на самообразование в области создания и использования управляемых сервисно-ориентированных приложений, способность самостоятельно освоить новые технологии, предоставляемые Web-ориентированными платформами.

Творческий уровень является показателем достижения высокой степени творческой активности. Специалисты готовы к широкому комплексному использованию в информационной деятельности технологий Web-ориентированных платформ, умеют самостоятельно ставить задачи, выбирать методы и средства для их решения, оценивать результаты применения выбранных методов. На этом уровне у обучаемых присутствует направленность на творческую деятельность – получение новой информации.

Отметим, что границы между уровнями достаточно условные, а комплекс знаний, умений и навыков изменяется в зависимости от степени информатизации образования.

Выявленные характеристики и компоненты современных Web-ориентированных платформ, а также основные направления обучения специалистов в области информатики созданию и использованию управляемых сервисно-ориентированных приложений позволили определить навыки и умения, необходимые для правильного и эффективного использования технологий Web-ориентированных платформ, в частности платформы .NET, которые соответствуют описанным уровням (таблица 7).

Таблица 7

**Уровни обученности в области использования технологий платформы .NET  
 при создании управляемых сервисно-ориентированных приложений**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Направления обучения**  **Уровни овладения способами инф. деят-ти** | Репродуктивный | Адаптивный | Эвристический | Творческий |
| Характерные осо-бенности объектно-ориентированного программирова-ния в среде .NET Framework | Использование основных классов, предоставляемых средой .NET Framework, необходимых для ввода и вывода информации.  Создание собственных простых классов | Комплексное исполь-зование разнообразных классов среды .NET Framework для создания эффективных алгоритмов | Доработка функциональности стандартных классов.  Использование технологий удаленного взаимодействия объектов. | Написание библиотек и компонент, состоящих из множества классов и имеющих сложные связи. |
| Инструментарий объектно-ориенти-рованного прог-раммирования в среде .NET Framework | Работа с редактором кода и использование справочной системы. | Настройка предлагаемых средств разработки. | Использование и настройка шаблонов проектов и утилит, входящих в среду разработки. Настройка среды разработки для создания групповых проектов.  Возможность удаленной отладки приложений.  Выбор и использование различных сред разработки. | Создание новых шаблонов проектов.  Создание собственных мастеров проектов.  Создание собственных утилит и интеграция их со средой разработки. |
| Разработка безопасных приложений | Умение различать типы угроз и уязвимостей программного обеспе-чения. | Использование средств безопасности, предостав-ляемых средой .NET Framework. | Использование технологий безопасного доступа кода и безопасности, основанной на ролях пользователей.  Обеспечение безопасности Web-приложений и XML Web-сервисов. | Разработка безопасных приложений, использующих возможности существующих серверов приложений. Эффективно использование алгоритмов криптографии. |
| Создание тради-ционных (консоль-ных и Windows) приложений | Использование потоков и файлов данных.  Использование технологии Windows Forms для создания простых приложений.  Технология использования различных окон диалога в приложениях. Разработка безопасных приложений. | Создание приложений, использующих возможности файловых систем.  Использование техники сериализации объектов. Создание и настройка простых форм.  Использование визуальных компонентов. | Создание многопоточных приложений.  Использование техники асинхронного программирова-ния.  Создание приложений, способ-ных обмениваться информа-цией, используя основные Интернет протоколы (HTTP, TCP, UDP).  Создание SDI- и MDI-приложений.  Использование визуального наследования.  Создание и использование справочных систем в разрабатываемых приложе-ниях.  Создание отчетов. | Использование технологии .NET Remoting.  Эффективное взаимодействие с неуправляемым кодом. Создание графических компонент, использующих технологию GDI+. |
| Создание Web-ориентированных приложений на базе платформы .NET | Создание и использование простых форм данных. | Использование серверных компонент. | Использование компонент проверки ввода пользователя.  Создание пользовательских элементов управления.  Управление состоянием.  Конфигурирование и развертывание Web-приложений. | Оптимизация и защита Web-приложений. |
| Технологии создания и использования баз данных на базе платформы .NET | Использование управляемых поставщиков данных. | Извлечение и обработка данных. | Использование класса DataSet. Установка связи с данными в компонентах.  Создание и использование хранимых процедур. Использование механизма транзакций.  Создание приложений, выполняющихся в присое-диненной и отсоединенной средах. | Проектирование баз данных. Защита информации в современных базах данных. |
| Создание сервисно-ориентированных приложений | Знание основных концепций сервисно-ориентированной архитектурой и сервисно-ориентированной технологии программирования.  Создание простых Web-сервисов. | Создание и настройка XML Web-сервисов. | Использование XML-технологий (XML, XSL, XSLT, Schema).  Использование WSDL и SOAP для настройки Web-сервисов. Использование основных Интернет-протоколов (HTTP, TCP, IP).  Публикация и внедрение XML Web-сервисов. | Проектирование XML Web-сервисов.  Защита XML Web-сервисов. |
| Проектирование приложений | Умение понять задачу и составить ТЗ. | Использование различных техник по сбору информации.  Составление плана и графика работ. | Использование модели MSF (Microsoft Solutions Framework) для разработки приложений. Выполнять различные роли модели MSF при разработке приложений. | Умение организовать процесс разработки сложных программных систем. |

Таким образом, будущие специалисты в области информатики должны владеть следующими умениями и навыками при создании и использовании управляемых сервисно-ориентированных приложений в соответствии с основными направлениями обучения:

1. При программировании в среде .NET Framework:

* изучать и анализировать возможности средств современных Web-ориентированных платформ и эффективно их использовать при создании приложений различного типа и назначения;
* знать и применять технологии объектно-ориентированного программирования при разработке приложений;
* описывать структуру, возможности и основные элементы платформы .NET;
* описать среду .NET Framework и ее компоненты;
* объяснять особенности использования различных языков программирования и их поддержку .NET Framework;
* описывать различия между ссылочными и размерными типами данных, используемые в среде .NET Framework;
* описывать основные пространства имен .NET Framework;
* использовать возможности развертывания и политики версий платформы .NET для развертывания различных программных компонентов;
* объяснять этапы компиляции и выполнения кода в управляемой среде;
* упаковывать и развертывать простые сборки, а также приложения, состоящие из множества сборок.

2. При изучении инструментальных средств создания управляемых сервисно-ориентированных приложений:

* иметь опыт использования среды программирования Visual Studio .NET при создании различных приложений;
* создавать сборки с сильным именем (strong-named assemblies);
* устанавливать сборки в глобальный кэш сборок (GAC);
* настраивать приложения .NET для управления местоположением и версиями его составных сборок;
* описывать преимущества использования сборки мусора;
* разрабатывать и настраивать распределенные приложения и используя технологию .NET Remoting.

3. При разработке безопасных приложений:

* создавать многопоточные приложения и управлять выполнением различных потоков;
* обеспечивать безопасное выполнение потоков в разрабатываемом приложении;
* использовать технологию Platform Invoke для вызова функций, находящихся в динамически подключаемых библиотеках (dll-файлах);
* использовать методы и свойства COM-компонентов в разрабатываемых сборках;
* применять различные методики обеспечения безопасности при создании приложений различного типа.

4. При создании традиционных (консольных и Windows) приложений:

* использовать технологию Platform Invoke для вызова функций, находящихся в динамически подключаемых библиотеках (dll-файлах);
* использовать методы и свойства COM-компонентов в разрабатываемых сборках;
* создавать Windows-приложения, используя технологию WinForms;
* использовать различные окна диалогов;
* проверять правильность пользовательского ввода;
* создавать и использовать пользовательские элементы управления;
* добавлять функции печати документов в создаваемые приложения;
* добавлять специальные возможности (accessibility features) в приложения;
* создавать локализованные версии разрабатываемых приложений;
* использовать справочные системы;

5. При создании Web-ориентированных приложений на базе платформы .NET:

* создавать Web-приложения, используя технологию ASP.NET;
* обращаться к Web-сервису из приложения ASP.NET и объединить полученные данные в Web-приложение;
* создавать приложения с сохранением состояния различными способами;
* настраивать и внедрять приложения ASP.NET;
* защищать приложения ASP.NET с использованием различных технологий защиты;

6. При создании сервисно-ориентированных приложений:

* объяснять сервисно-ориентированный подход к созданию информационных систем и его основные элементы;
* создавать и использовать XML Web-сервисы.

7. При использовании технологий баз данных:

* уметь проектировать и использовать базы данных;
* описывать и использовать объектную модель технологии ADO .NET;
* извлекать данные используя специальные классы;
* составлять структурированные запросы к информационным ресурсам распределенных баз данных;
* уметь проектировать и использовать базы данных.

8. При проектировании приложений специалист должен уметь использовать модель MSF для организации групповой работы.

Таким образом, для оценки результатов подготовки будущих специалистов в области информатики в области создания и использования управляемых сервисно-ориентированных приложений определены четыре уровня обученности. На репродуктивном уровне обучаемые лишь воспроизводят способы информационной деятельности. При достижении адаптивного уровня они способны применить усвоенные способы в новой ситуации. Эвристический уровень показывает, что будущие специалисты в области информатики способны самостоятельно использовать технологии современных Web-ориентированных платформ для решения поставленных задач, правильно выбрав способы их применения. Творческий уровень означает широкое комплексное использование в информационной деятельности будущих специалистов средств разработки управляемых сервисно-ориентированных приложений, самостоятельную постановку задач, выбор методов и средств для их решения.

**Библиография**

1. Балл Г.А. Теория учебных задач.- М.: Педагогика, 1990. – 1984 с.
2. Беспалько В.П. Педагогика и прогрессивные технологии обучения. – М., 1995. –336 с.
3. Гальперин П.Я. Основные результаты исследований по проблеме “Формирование умственных действий и понятий”. М., 1965.
4. Лернер И.Я. Дидактические основы методов обучения. ‑ М., 1981. ‑ С. 29.
5. Шухман А.Е. Совершенствование содержания подготовки педагогических кадров к применению информационных и коммуникационных технологий в профессиональной деятельности: Дисс. ... канд. пед. наук. ‑ М., 2000. ‑ 149 с.
6. Эльконин Д.Б. Избранные педагогические труды. – М.: Педагогика, 1989.
7. Талызина Н.Ф. Теоретические основы разработки модели специалиста. – М., 1986.
8. Пахомова Н.Ю. Развитие методики использования “учебных проектов” при обучении информатике в общеобразовательной школе. Авторф. дис. канд.пед.наук. – М., 1997. – 19 с.

**---------------------------------------------------------------------------------------------------------------------**

Российский портал информатизации образования [содержит: законодательные и нормативные правовые акты государственного регулирования информатизации образования, федеральные и региональные программы информатизации сферы образования, понятийный аппарат информатизации образования, библиографию по проблемам информатизации образования, по учебникам дисциплин цикла Информатика, научно-популярные, документальные видео материалы и фильмы, периодические издания по информатизации образования и многое другое.](http://portalsga.ru)

