

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Саратовский государственный технический университет
имени Гагарина Ю.А.»

Кафедра «Информационная безопасность автоматизированных систем»

РАБОЧАЯ ПРОГРАММА

по дисциплине

Б.3.3.2.2 «Технология разработки программного обеспечения»

направления подготовки

10.03.01 "Информационная безопасность"

Профиль «Безопасность автоматизированных систем»

форма обучения – очная

курс – 2

семестр – 3

зачетных единиц – 4,5

часов в неделю – 5,5

всего часов – 180

в том числе:

лекции – 32

лабораторные занятия – 36

коллоквиумы - 4

самостоятельная работа – 108

экзамен – 4 семестр

1. Цели и задачи дисциплины

Цель преподавания дисциплины: является изучение современных технологий и методов программирования, получение навыков проектирования и разработки программного обеспечения (ПО), расширение кругозора в сфере разработки ПО.

Задачи изучения дисциплины:

- изучение методов проектирования ПО;
- изучение оценки качества программного обеспечения;
- изучение тестирования и отладки программного обеспечения;
- изучение принципов, методов и средств сопровождения ПО;
- изучение структур данных;
- изучение алгоритмов и навыков их практической реализации при разработке программных систем.
- изучение методологии и средств разработки ПО;
- умение грамотно пользоваться языком предметной области
- знание методов проектирования и производства программного продукта, принципов построения, структуры и приемов работы с инструментальными средствами, поддерживающими создание программного обеспечения
- формирования навыков выбора, проектирования, реализации, оценки качества и анализа эффективности программного обеспечения для решения задач в различных предметных областях

2. Место дисциплины в структуре ООП ВПО

Дисциплина «Технология разработки программного обеспечения» относится к числу дисциплин по выбору базовой части профессионального цикла.

Для успешного усвоения данной дисциплины необходимо, чтобы студент владел знаниями, умениями и навыками, сформированными в процессе изучения дисциплин:

«Информатика» - знать формы и способы представления данных в персональном компьютере, классификацию современных компьютерных систем, типовые структуры и принципы организации компьютерных сетей; уметь применять типовые программные средства сервисного назначения (средства восстановления системы после сбоев, дефрагментации и очистки диска и т.п.), пользоваться сетевыми средствами и внешними носителями информации для обмена данными; владеть навыками обеспечения безопасности информации с помощью типовых программных средств, навыками поиска и обмена информацией в глобальной сети Интернет;

Дисциплина «Технологии и методы программирования» является предшествующей для изучения следующих базовых дисциплин: «Разработка и эксплуатация защищенных автоматизированных систем», «Программно-аппаратные средства обеспечения информационной безопасности», «Организация ЭВМ и вычислительных систем».

3. Требования к результатам освоения дисциплины

Изучение дисциплины направлено на формирование следующих компетенций:

способностью к обобщению, анализу, восприятию информации, постановке цели и выбору путей ее достижения, владеть культурой мышления (ОК-8);

способностью к саморазвитию, самореализации, приобретению новых знаний, повышению своей квалификации и мастерства (ОК-11);

способностью использовать инструментальные средства и системы программирования для решения профессиональных задач (ПК-16);

способностью применять методы анализа изучаемых явлений, процессов и проектных решений (ПК-20);

Студент должен знать:

- современные подходы к разработке программного обеспечения
- принципы работы интернет-приложений;
- основные принципы кросс-браузерной верстки;
- принципы обмена данными между клиентами и сервером в сети Internet:
 - правила адресации устройств в сети;
 - протокол HTTP;
- форматы передачи данных:
 - JSON;
 - XML;

Студент должен уметь:

- профессионально выстраивать стратегию разработки и реализации приложений;
- планировать архитектуру приложений с учетом современных тенденций разработки;
- выявлять общие характеристики у разнотипных данных и обрабатывать их с применением принципов объектно-ориентированного проектирования;

Студент должен владеть:

- навыками практического использования алгоритмов и структур данных при решении конкретных задач;

- навыками практического применения концепций объектно-ориентированного программирования;
- практической работы по созданию и разработке web-приложений

4. Распределение трудоемкости (час.) дисциплины по темам и видам занятий

№ модуля	№ недели	№ темы	Наименование темы	Часы/ Из них в интерактивной форме					
				Всего	Лекции	Коллоквиумы	Лабораторные	Практические	СРС
1	2	3	4	5	6	7	8	9	10
1	1	1	Advanced JavaScript	-	6	-	-	-	8
1	4	2	ООП в JavaScript	-	6	-	10	-	14
1	7	3	UnderscoreJS & LodashJS Шаблоны	-	6	-	-	-	10
1	10	4	Протокол HTTP	-	2	-	-	-	14
2	11	5	Введение в AngularJS	-	2	-	-	-	16
2	12	6	Анатомия AngularJS	-	6	-	-	-	18
2	15	7	AngularJS Controllers and Scopes	-	4	-	10	-	14
2	17	8	AngularJS Built-in directives	-	-	4	16	-	14
Всего				360	32	4	36	-	108

5. Содержание лекционного курса

№ темы	Всего часов	№ лекции	Тема лекции. Вопросы, отрабатываемые на лекции	Учебно-методическое обеспечение
1	2	3	4	5
1	2	1	Тема 1. Advanced JavaScript. Типы данных: [[Class]], instanceof, формат JSON, метод toJSON, setTimeout и setInterval, запуск кода из строки: eval, перехват ошибок, «try..catch»	1-9
2	2	2	Глобальный объект, замыкания, функции изнутри [[Scope]] для new Function, локальные переменные для объекта, модули через замыкания, управление памятью в JavaScript, устаревшая конструкция «with»	1-9
2	2	3	Методы объектов, this, преобразование объектов: toString и valueOf, создание объектов через «new», дескрипторы, геттеры и сеттеры свойств, статические и фабричные методы, явное указание this: «call», «apply», привязка контекста и карринг: «bind»,	1-9

			функции-обёртки, декораторы.	
2	2	4	Тема 2. ООП в JavaScript Введение, внутренний и внешний интерфейс, геттеры и сеттеры, функциональное наследование	1-9
3	2	5	Прототип объекта, свойство F.prototype и создание объектов через new, встроенные «классы» в JavaScript, свои классы на прототипах	1-9
4	2	6	Наследование классов в JavaScript, проверка класса: «instanceof», свои ошибки, наследование от Error, примеси	1-9
5	2	7	Тема 3. UnderscoreJS & LodashJS. Введение, шаблоны, коллекции, массивы, функции.	1-9
5	2	8	Объекты, утилиты, цепные вызовы.	1-9
6	2	9	Индивидуальные сборки, функции для работы с объектами, ООП обертки и цепочки.	1-9
7	2	10	Тема 4. Протокол HTTP. Виды HTTP запросов, заголовки, тело сообщения, REST.	1-9
7	2	11	Тема 5. Введение в AngularJS. Введение, паттерн MVC, первое приложение, представление, контроллер, модель.	1-9
8	2	12	Тема 6. Анатомия AngularJS. Директивы, выражение, формы, локализация, модули области видимости.	1-9
8	2	13	Внедрение зависимостей, сервис, фабрика, провайдер, сервисы в AngularJS.	1-9
9	2	14	Модули, фильтры, создание и применение фильтров.	1-9
9	2	15	Тема 7. AngularJS Контролеры и области данных. Зависимости в контролерах, REST сервисы, наследование областей видимости.	1-9
10	2	16	Тестирование контроллеров, сервисы в контроллерах, минификация.	1-9

6. Содержание коллоквиумов

Коллоквиумы учебным планом не предусмотрены.

7. Перечень практических занятий

№ темы	Всего часов	№ занятия	Тема практического занятия. Задания, вопросы, отрабатываемые на практическом занятии	Учебно-методическое обеспечение
1	2	3	4	5
1	2	1	Advanced JavaScript	1-9
2	4	2	ООП в JavaScript	1-9
3	2	4	UnderscoreJS & LodashJS Шаблоны	1-9
4	2	5	Протокол HTTP	1-9
5	2	6	Введение в AngularJS	1-9
6	4	7	Анатомия AngularJS	1-9
7	2	8	AngularJS Controllers and Scopes	1-9

8. Перечень лабораторных работ

№ темы	Всего часов	Наименование лабораторной работы. Задания, вопросы, отрабатываемые на лабораторном занятии	Учебно-методическое обеспечение
1	2	4	3
2	10	ООП в JavaScript	1-9
7	10	AngularJS Controllers and Scopes	1-9
8	16	AngularJS Built-in directives	1-9

8. Задания для самостоятельной работы студентов

№ темы	Всего Часов	Задания, вопросы, для самостоятельного изучения (задания)	Учебно-методическое обеспечение
1	2	3	4
1	6	Advanced JavaScript	1-9
2	10	ООП в JavaScript	1-9
3	8	UnderscoreJS & LodashJS Шаблоны	1-9
4	12	Протокол HTTP	1-9
5	14	Введение в AngularJS	1-9
6	16	Анатомия AngularJS	1-9
7	12	AngularJS Controllers and Scopes	1-9
8	12	AngularJS Built-in directives	1-9

Виды, график контроля СРС, (по решению кафедры УМКС/УМКН).

№ темы	Вид СРС	Вид контроля СРС	График контроля (№ недели)
1-5	Работа с печатными источниками, разбор типовых заданий	Рубежный контроль, промежуточный контроль, самоконтроль	8 (промежуточная аттестация), экзамен
6-11	Работа с печатными источниками, разбор типовых заданий	Рубежный контроль, промежуточный контроль, самоконтроль	Экзамен

10. Расчетно-графическая работа

Расчетно-графическая работа учебным планом не предусмотрена.

11. Курсовая работа

Курсовая работа учебным планом не предусмотрена.

12. Курсовой проект

Курсовой проект учебным планом не предусмотрен.

13. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине (модулю)

Формирование профессиональных компетенций по дисциплине производится на лабораторных и лекционных занятиях (75%); закрепление достигается при проведении промежуточной аттестации (10%) и сдаче экзамена (15%).

Вопросы для зачета

Зачет учебным планом не предусмотрен.

Вопросы для экзамена

1. Scope в javascript
2. Специфика this
3. Исключения и перехват ошибок в JavaScript
4. всплытие события
5. Замыкания
6. Методы call и apply
7. Что такое прототипное наследование
8. Инкапсуляция в JavaScript
9. Что такое Lodash и Underscore. Когда какой использовать.
10. HTTP и виды запросов.
11. Что такое REST.
12. Подключение и использование AngularJS
13. Контроллер и область видимости AngularJS.
14. Сервисы в AngularJS.
15. Модули и зависимости в AngularJS.
16. Стандартные директивы AngularJS.
17. Создание собственных директив AngularJS.
18. Роутинг в AngularJS.
19. Архитектура AngularJS приложения.
20. Тестирование AngularJS.

Тестовые задания по дисциплине

Для проведения тестирования используются тестовые материалы, разработанные в среде АСТ-Тест.

14. Образовательные технологии

Для реализации компетентностного подхода в соответствии с требованиями ФГОС ВО в рамках учебного курса предусмотрены активные и интерактивные формы проведения занятий в сочетании с внеаудиторной работой с целью формирования и развития профессиональных навыков обучающихся.

В связи с этим предусмотрено применение мультимедийных средств и презентаций, обсуждение докладов студентов, лекции с элементами деловых игр, тестирование, консультации, решение ситуационных задач, дискуссии.

Общее количество занятий, проводимых в интерактивных формах, не менее 24 часов.

15. ПЕРЕЧЕНЬ УЧЕБНО-МЕТОДИЧЕСКОГО ОБЕСПЕЧЕНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

Обязательные издания

1. Антонова, Г. М. Современные средства ЭВМ и телекоммуникаций : учеб. пособие / Г. М. Антонова, А. Ю. Байков. - М. : ИЦ "Академия", 2010. - 144 с.;
2. Ломов, А. Ю. HTML, CSS, скрипты: практика создания сайтов / А. Ю. Ломов. - СПб. : БХВ-Петербург, 2006. - 416 с.
3. Прохоренок, Н. А. HTML, JavaScript, PHP и MySQL. Джентльменский набор Web-мастера [+CD] / Н. А. Прохоренок. - 3-е изд. - СПб. : БХВ-Петербург, 2012. - 912 с.

Дополнительные издания

4. Фримен А. ASP.NET MVC 4 с примерами на С# 5.0 для профессионалов / А.Фримен. - М.: Вильямс, 2013. - 688 с.
5. Чедвик Джесс ASP.NET MVC 4 разработка реальных веб-приложений с помощью ASP.NET MVC / Д.Чедвик, Т.Снайдер – М.: Вильямс, 2013 – 432 с.
6. Флэнаган Д. JavaScript Подробное руководство / Д. Флэнаган - М.: Символ-Плюс, 2008 – 992 с.
7. Freeman Adam PRO Angular JS / A. Freeman – NY.: Apress, 2014 –688 с.

Интернет-ресурсы

8. ASP.NET http://professorweb.ru/my/ASP_NET/base/level1/aspnet_info.php
9. Онлайн-книга 'Изучаем ASP.NET MVC 4'. <http://metanit.com/sharp/mvc/>

16. Материально-техническое обеспечение дисциплины.

Для проведения лекционных занятий требуется комплект технических средств обучения в составе:

- персональный компьютер (в конфигурации не хуже: процессор Intel Core 2 Duo, 2 Гбайта ОЗУ, 500 Гбайт НЖМД);
- проектор (разрешение не менее 1280x1024);
- экран для проектора.

Для проведения практических занятий требуется компьютерный класс, оборудованный ПЭВМ в конфигурации не худшей чем: процессор Pentium IV 3 ГГц, ОЗУ 2 Гбайта, НЖМД 200 Гбайт.