

Аннотация рабочей программы дисциплины

Направление подготовки: 02.03.02 «Фундаментальная информатика и информационные технологии»

Тип образовательной программы: академический бакалавриат

Направленность (профиль): Информационная сфера

Квалификация (степень) выпускника: бакалавр

Форма обучения: очная

1. Наименование дисциплины

Б1.В.ДВ.1.2 Технологии программирования

2. Цели и задачи дисциплины (модуля):

Цель дисциплины - освоение студентами практических навыков в области разработки программного обеспечения на основе современных подходов к проектированию сложных, гетерогенных, распределенных информационных систем.

Задачи дисциплины - изложение основных положений концепции компонентного проектирования, экономических аспектов его использования на предприятии, а также усвоение приемов программирования, широко используемых в этой области.

3. Требования к результатам освоения дисциплины (модуля):

Процесс изучения дисциплины направлен на формирование следующих компетенций: ОК-7 (способность к самоорганизации и самообразованию), ПК-2 (способность понимать, совершенствовать и применять современный математический аппарат, фундаментальные концепции и системные методологии, международные и профессиональные стандарты в области информационных технологий).

В результате изучения дисциплины студент должен:

Знать:

- основные фундаментальные понятия, лежащие в основе компонентного подхода к проектированию программного обеспечения;
- методы интерпретации процедур системного анализа в виде этапов проектирования сложных программных систем;
- методы моделирования компонентных систем при помощи средств языков UML и BPMN;
- методики проектирования интерфейсов компонент;
- методики проектирования реализаций компонент;
- Технологии и языки задания конфигураций программных модулей.

Уметь:

- проектировать программные системы при помощи современных сред компонентного проектирования;
- создавать программные пакеты, регистрировать их на серверах;
- разрабатывать тестовые компоненты, анализировать результаты тестов;
- анализировать критерии качества проектирования компонент и их интерфейсов.

Владеть:

- средами программирования, поддерживающими компонентный подход;
- средствами самостоятельной разработки среды для используемых языков программирования;
- программными средствами доступа к хранилищам программных компонент и их загрузке;
- методиками экстремального и *agile*-программирования.

4. Объем дисциплины (модуля) и виды учебной работы

Вид учебной работы	Всего часов / зачетных	Семестры			
		4			

	единиц				
Аудиторные занятия (всего)	40	40			
В том числе:	-	-	-	-	-
Практические занятия (ПЗ)	36	36			
Самостоятельная работа (всего)	32	32			
В том числе:	-	-	-	-	-
<i>Другие виды самостоятельной работы</i>	32	32			
Вид промежуточной аттестации	Зачет	Зачет			
Контактная работа (всего)	43	43			
Общая трудоемкость	часы	72	72		
	зачетные единицы	2	2		

5. Краткая характеристика содержания учебной дисциплины

Тема 1. Введение в компонентное программирование программного обеспечения.

Парадигма компонентного программирования: основные положения. Основные понятия: «интерфейс», «компонента», «обслуживание», «реализация», «утилита», «адаптер», «утилита» (сервис), «контейнер», «фабрика», «шаблон проектирования» и др. Обзор популярных сред компонентного проектирования. Проектирование интерфейсов компонент.

Тема 2. Адаптер-ориентированное программирование.

Понятие адаптера. Динамическая и статическая адаптация. Виды адаптеров: прокси-адаптеры, фасады, MVW и т.п. Приемы проектирования компонентных пакетов (библиотек). Разновидности адаптеров: Proxy-адаптеры, фасады, фабрики, композиты, мосты и др. Динамическая и статическая адаптация.

Тема 3. Методики конфигурирования компонентного приложения.

Конфигурирование компонентной программной системы. Виды конфигураций, подходы к созданию конфигураций: императивный и декларативный. Язык ZCML (ZOPE Configuration Markup Language). Структура INI-файлов. Применение INI-файлов для конфигурирования индивидуальных утилит.

Тема 4. Компоненты WEB.

Современные подходы к проектированию web-приложений и их реализации в среде Python: flask, pyramid, ZOPE, Django и др. Отображение запросов HTTP на процедуры и методы. Моделирование шаблонов проектирования при реализации web-приложений. Возможности представления графики в web. Шаблоны Модель-Вид-Контроллер (Model-View-Controller), Модель-Вид-Презентер (Model-View-Presenter), Модель-Вид-ВидМодель (Model-View-ViewModel), и др. Отображение таблиц баз данных на модели MVW (Model-View-"Whatever").

Тема 5. Распределенные программные системы.

Понятие API (Application Programming Interface) сервисов и их отличия от API библиотек и пакетов. Интерфейсы сервисов. RESTful-сервисы. Пакеты реализации RESTful-сервисов в среде программирования Python. Методы кодирования информации уровня представления данных модели OSI/ISO. Форматы XML, JSON, BSON, байтовые потоки и WebSocket-ы. Протоколы RPC и XML-RPC.

Тема 6. Реализация порождающего программирования.

Шаблоны в языках программирования C++, Java, C#, D, Vala, Julia. Обобщенное программирование и его реализация в перечисленных системах. Методики конфигурирования для систем порождающего программирования данного класса (Чернецки).

Тема 7. Автоматизация обработки спецификаций программного обеспечения.

Основы MDE (Model Driven Engineering)-программирования, основанного на преобразовании модели программного обеспечения. Спецификации программного обеспечения, представляемые в виде моделей. Виды MDE: DDE (Data Driven Engineering), MDA (Model Driven Engineering), ODE (Ontology Driven Engineering). MDE как процесс моделирования предметной области. Фундаментальные проблемы технологий MDE.

6. Форма промежуточной аттестации:

зачет

7. Разработчик аннотации

Зав. кафедрой информационных технологий А.Е. Хмельнов