

**Федеральное государственное автономное образовательное
учреждение высшего образования
«Московский физико-технический институт
(национальный исследовательский университет)»**

УТВЕРЖДЕНО
Директор

А.В. Малеев

	Рабочая программа дисциплины (модуля)
по дисциплине:	Архитектура и дизайн ПО
по направлению:	Программная инженерия
профиль подготовки:	Разработка программно-информационных систем высшая школа программной инженерии высшая школа программной инженерии
курс:	3
квалификация:	бакалавр

Семестр, формы промежуточной аттестации: 6 (весенний) - Экзамен

Аудиторных часов: 60 всего, в том числе:

лекции: 30 час.

семинары: 30 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 18 час.

Подготовка к экзамену: 30 час.

Всего часов: 108, всего зач. ед.: 3

Программу составил: А.В. Созыкин, канд. техн. наук

Программа обсуждена на заседании высшей школы программной инженерии 13.05.2024

Аннотация

Дисциплина «Архитектура, дизайн и процесс разработки ПО» направлена на получение теоретических знаний о современных методах и подходах к построению сервисов. По ходу освоения программы студенты осваивают базу знаний систем, используемых в современных продуктах и получают навык разработки прикладных программ. Помимо теоретических знаний студенты получают знания об основных методологиях разработки и паттернах проектирования программного обеспечения.

В дисциплине рассматриваются следующие темы:

- Процесс разработки программного обеспечения.
- Системы контроля версий.
- Базовые навыки работы с *nix системами.
- Базовый обзор информационной безопасности.
- Базовое введение в компьютерные сети.
- Базы данных.
- Стек протоколов TCP/IP, HTTP.
- Типовые архитектуры веб-приложений.
- Паттерны проектирования GoF.
- Инструменты прототипирования.

Заключительным этапом курса является экзамен, который имеет целью проверку знаний студентов по теории и выявления практических навыков применения полученных знаний при разработке современных сервисов.

1. Цели и задачи

Цель дисциплины

- дать студентам представление о современных методах и подходах к построению сервисов, дать обзорную базу знаний систем, используемых в современных продуктах, развить навык разработки прикладных программ.

Задачи дисциплины

- овладеть основными методологиями построения процессов разработки;
- овладеть базовыми знаниями работы с *nix системами;
- получить общее представление о компьютерных сетях, базах данных, криптографии;
- дать объяснение основным паттернам проектирования и инструментам прототипирования.

2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
ОПК-2 Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности	ОПК-2.1 Способен использовать информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности
	ОПК-2.2 Владеет навыками создания программного обеспечения для ЭВМ и систем различной архитектуры
ОПК-5 Способен устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем	ОПК-5.2 Способен оценивать концепции и атрибуты качества программного обеспечения (надежности, безопасности, удобства использования), в том числе роль людей, процессов, методов, инструментов и технологий обеспечения качества
	ОПК-5.3 Владеет навыками установки программного и аппаратного обеспечения информационных и автоматизированных систем
ПК-5 Способен проектировать,	ПК-5.1 Умеет описывать архитектуру, устройство и функционирование информационных систем

разрабатывать, внедрять, сопровождать и снимать с эксплуатации информационные системы	ПК-5.2 Умеет определять оптимальные методы и инструменты разработки, внедрения и адаптации прикладного программного обеспечения информационных систем
ПК-6 Способен разрабатывать и внедрять стандарты и процессы разработки, производить их мониторинг и обновления	ПК-6.3 Владеет навыками мониторинга и обновления стандартов с учетом изменяющихся требований и технологий

3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны

знать:

- основные методологии разработки программного обеспечения;
- основные паттерны проектирования;
- на базовом уровне организацию компьютерных сетей, стек протоколов TCP/IP.

уметь:

- работать с *nix системами;
- обращаться с базами данных и минимальными инструментами анализа данных.

владеть:

- навыком работы с *nix системами;
- инструментами прототипирования.

4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

4.1. Разделы дисциплины (модуля) и трудоемкости по видам учебных занятий

№	Тема (раздел) дисциплины	Трудоемкость по видам учебных занятий, включая самостоятельную работу, час.			
		Лекции	Семинары	Лаборат. работы	Самост. работа
1	Процесс разработки программного обеспечения.	5	5		2
2	Системы контроля версий.	5	5		2
3	Базовые навыки работы с *nix системами.	2	2		1
4	Базовый обзор информационной безопасности.	2	2		1
5	Базовое введение в компьютерные сети.	1	1		1
6	Базы данных.	5	5		3
7	Стек протоколов TCP/IP, HTTP.	5	5		3
8	Типовые архитектуры веб-приложений.	2	2		2
9	Паттерны проектирования GoF.	2	2		2
10	Инструменты прототипирования.	1	1		1
Итого часов		30	30		18
Подготовка к экзамену		30 час.			
Общая трудоёмкость		108 час., 3 зач.ед.			

4.2. Содержание дисциплины (модуля), структурированное по темам (разделам)

1. Процесс разработки программного обеспечения.

Ключевые процессы: анализ, проектирование, программирование, документирование, тестирование. Жизненный цикл проекта. Модели: водопадная модель (каскадная, последовательная), итерационная модель, спиральная модель. Гибкие методологии: Agile (Lean, Scrum, FDD и др.), RUP, MSF, DSDM. Практики программирования: парное программирование, непрерывная интеграция, разработка через тестирование. PMBoK.

2. Системы контроля версий.

Централизованные, распределенные. ПО с открытым исходным кодом, сообщество вокруг него.

3. Базовые навыки работы с *nix системами.

Необходимо овладеть следующими базовыми навыками:

- устанавливать nix в графическом режиме;
- управлять физическими хранилищами данных;
- устанавливать и настраивать локальные компоненты и службы;
- настраивать сетевые соединения и безопасность сетевых служб;
- управлять файлами и обеспечивать их безопасность;
- администрировать пользователей и группы;
- разворачивать службы общего доступа к файлам.

4. Базовый обзор информационной безопасности.

Типы уязвимостей, шифрование.

5. Базовое введение в компьютерные сети.

Принципы формирования и типы сетей. Назначение компьютерных сетей. Основные программные и аппаратные компоненты сети. Классификация компьютерных сетей.

6. Базы данных.

История. Виды баз данных. Классификация по модели данных. Классификация по среде постоянного хранения. Классификация по содержанию. Классификация по степени распределённости. Другие виды БД.

7. Стек протоколов TCP/IP, HTTP.

Стандартные стеки коммуникационных протоколов.

OSI.

TCP/IP.

IPX/SPX.

NetBIOS/SMB.

8. Типовые архитектуры веб-приложений.

Распределение нагрузки. Высокая доступность. Типовые архитектуры мобильных приложений.

9. Паттерны проектирования GoF.

Порождающие шаблоны проектирования. Поведенческие шаблоны проектирования. Структурные шаблоны проектирования.

10. Инструменты прототипирования.

Axure RP Pro Инструмент, ориентированный на создание прототипов веб-сайтов. Генерирует кликабельный HTML и документацию в формате Word. Поддерживает комплексное взаимодействие. Windows

Balsamiq Mockups позволяет очень быстро создавать макеты вашего ПО. Сгенерированное содержимое выглядит как скетчи,

CogTool* Создаёт простые макеты пользовательского интерфейса и позволяет оценить их эффективность.

Coutline* Веб-приложение для создания и просмотра интерактивных прототипов.

Dreamweaver Используйте визуальную часть Dreamweaver для перетаскивания и размещения элементов дизайна с помощью drag-and-drop, добавления элементов интерактивности, и погружайтесь в код для более комплексного прототипирования. Кроссплатформенный

EasyPrototype* Очень похож на популярный Axure, легкий инструмент, позволяет проектировать экранные формы, экспортировать интерактивные HTML-прототипы и документацию. Кроссплатформенный

Excel*

Expression Blend генерирует прототипы для Silverlight и WPF приложений с богатыми интерактивными возможностями,

Expression Blend + SketchFlow* Создание карт потока задач и концепций интерфейсов, которые выглядят как скетчи.

Expression Design Мощный инструмент рисования для создания прототипов HTML, Silverlight и WPF приложений с ограниченной интерактивностью.

Fireworks возможно создание сложных интерактивных прототипов. Множество инструментов аналогичны некоторым инструментам из Adobe suite.

FlairBuilder* Создаёт интерактивные экранные формы с помощью десктопного Air приложения

Flash быстро генерирует анимацию или простые интерактивные прототипы

Flash Catalyst Инструмент, еще находящийся в процессе разработки, призван помочь дизайнерам в создании интерфейсов для флэш-приложений. Кроссплатформенный

Flex несмотря на то, что более приспособлен для разработчиков, WYSIWYG редактор и поддержка импорта скинов

ForeUI* Создаёт макеты, определяет и моделирует поведение приложения в браузере. Кроссплатформенный

FormBuilder for Drupal имеет веб-интерфейс с возможностью перетаскивания элементов на страницу.

GUI Design Studio* Создаёт интерфейсы, аннотации к ним, строит раскладовки для определения рабочего прототипа.

iPlotz* Веб-приложение, создающее интерактивные экранные формы.

iRise Комплексный инструмент для моделирования бизнес-процессов и проектирования интерфейса приложения.

Justinmind Prototyper* Создаёт экранные формы с возможностью определения их поведения через описание с помощью use case-диаграмм. Кроссплатформенный

JustProto* Веб-приложение, ориентированное на работу с удалённой командой

Keynote Похож на Powerpoint.

LiveView Просмотр вашего рабочего стола на виртуальном iPhone, или в качестве приложения на реальном iPhone.

Lucid Spec* Дизайн экранных форм и моделирование рабочих приложений.

5. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

Учебная аудитория, оснащенная компьютером и мультимедийным оборудованием (проектор, звуковая система). Во время лекций используются демонстрации презентаций совместно с прямой трансляцией экрана преподавателя для проведения практических демонстраций.

6.Перечень рекомендуемой литературы

Основная литература

1. Технологии разработки программного обеспечения, учебник для вузов, стандарт третьего поколения / С. А. Орлов, Б. Я. Цилькер. — Санкт-Петербург, Питер, 2012.— URL: <https://ibooks.ru/bookshelf/28460/reading> (дата обращения: 25.11.2020). - Полный текст (Режим доступа : из сети МФТИ / Удаленный доступ)

Дополнительная литература

Рекомендуемая литература для самостоятельного изучения:

Фримен, Эр. Паттерны проектирования : [учеб. пособие для вузов] / Эр. Фримен, Эл. Фримен ; при участии К. Сьерра, Б. Бейтса .— Санкт-Петербург : Питер, 2017 .— Электрон. версия
Архитектура компьютера и проектирование компьютерных систем [Текст] : [учеб. пособие для вузов] / Д. Паттерсон, Дж. Хеннесси ; [пер. с англ. Н. Вильчинский] .— 4-е изд. — СПб. : Питер, 2012 .— 784 с.

7. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)

Не используются

8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень необходимого программного обеспечения и информационных справочных систем (при необходимости)

На лекционных занятиях используются мультимедийные технологии, включая демонстрацию презентаций. Для контроля и коррекции знаний, обучающиеся могут использовать компьютерное тестирование.

9. Методические указания для обучающихся по освоению дисциплины (модуля)

Методические рекомендации позволяют студенту оптимальным образом организовать процесс обучения. В структуре учебного плана значительное время отводится на самостоятельное изучение данной дисциплины. В рабочей программе приведено примерное распределение часов аудиторной и внеаудиторной нагрузки по различным темам данной дисциплины.

Для успешного освоения данной дисциплины студенту необходимо:

- посещать лекции и семинары, при этом конспектирование материалов не является необходимым, поскольку основные материалы будут доступны в облачном хранилище wiki, к которому предоставлен доступ всем студентам кафедры;
- выполнять задания, задаваемые преподавателем на лекциях и семинарах;
- сдавать проверочные контрольные работы.

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

по направлению: Программная инженерия
профиль подготовки: Разработка программно-информационных систем
высшая школа программной инженерии
высшая школа программной инженерии
курс: 3
квалификация: бакалавр

Семестр, формы промежуточной аттестации: 6 (весенний) - Экзамен

Разработчик: А.В. Созыкин, канд. техн. наук

1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
ОПК-2 Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности	ОПК-2.1 Способен использовать информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности
	ОПК-2.2 Владеет навыками создания программного обеспечения для ЭВМ и систем различной архитектуры
ОПК-5 Способен устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем	ОПК-5.2 Способен оценивать концепции и атрибуты качества программного обеспечения (надежности, безопасности, удобства использования), в том числе роль людей, процессов, методов, инструментов и технологий обеспечения качества
	ОПК-5.3 Владеет навыками инсталляции программного и аппаратного обеспечения информационных и автоматизированных систем
ПК-5 Способен проектировать, разрабатывать, внедрять, сопровождать и снимать с эксплуатации информационные системы	ПК-5.1 Умеет описывать архитектуру, устройство и функционирование информационных систем
	ПК-5.2 Умеет определять оптимальные методы и инструменты разработки, внедрения и адаптации прикладного программного обеспечения информационных систем
ПК-6 Способен разрабатывать и внедрять стандарты и процессы разработки, производить их мониторинг и обновления	ПК-6.3 Владеет навыками мониторинга и обновления стандартов с учетом изменяющихся требований и технологий

2. Показатели оценивания компетенций

В результате изучения дисциплины «Архитектура и дизайн ПО» обучающийся должен:

знать:

- основные методологии разработки программного обеспечения;
- основные паттерны проектирования;
- на базовом уровне организацию компьютерных сетей, стек протоколов TCP/IP.

уметь:

- работать с *nix системами;
- обращаться с базами данных и минимальными инструментами анализа данных.

владеть:

- навыком работы с *nix системами;
- инструментами прототипирования.

3. Перечень типовых (примерных) вопросов, заданий, тем для подготовки к текущему контролю

Обучающиеся получают задание по освещению определённых теоретических вопросов или решению задач. Работа выполняется письменно дома и сдаётся преподавателю. Оцениваются владение материалом по теме работы, аналитические способности, владение методами, умения и навыки, необходимые для выполнения заданий.

Примерные теоретические вопросы:

1. Что такое программное обеспечение?
2. Жизненный цикл программного обеспечения.
3. Модели разработки программного обеспечения
4. Объектно-ориентированный подход.

5. Модель «водопада» разработки программного обеспечения.
6. Сервисно-ориентированное ПО.
7. Принцип открытости закрытости ПО.
8. Проектирование классов программного обеспечения.
9. Проектирование связей между классами программного обеспечения.
10. Уточнение отношений между классами, выявленными на этапе анализа.
11. Выделение подсистем на этапе проектирования.
12. Шаблоны архитектуры программного обеспечения.
13. Интерфейсы и компоненты.

4. Перечень типовых (примерных) вопросов и тем для проведения промежуточной аттестации обучающихся

1. Процесс разработки программного обеспечения. Ключевые процессы: анализ, проектирование, программирование, документирование, тестирование.
2. Системы контроля версий. Централизованные, распределенные. ПО с открытым исходным кодом, сообщество вокруг него.
3. Базовые навыки работы с *nix системами.
4. Базовый обзор информационной безопасности: типы уязвимостей, шифрование.
5. Базовое введение в компьютерные сети.
6. Базы данных.
7. стек протоколов TCP/IP, HTTP.
8. Типовые архитектуры веб-приложений. Распределение нагрузки. Высокая доступность. Типовые архитектуры мобильных приложений.
9. Паттерны проектирования GoF.
10. Инструменты прототипирования.

Билет 1

ПО с открытым исходным кодом, сообщество вокруг него.
Паттерны проектирования GoF.

Билет 2

Базовое введение в компьютерные сети.
Гибкие методологии: Agile (Lean, Scrum, FDD и др.), RUP, MSF, DSDM.

Критерии оценивания

- оценка «отлично (10)» выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений
- оценка «отлично (9)» выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений
- оценка «отлично (8)» выставляется студенту, показавшему всесторонние систематизированные, глубокие знания учебной программы дисциплины и умение применять их на практике при решении конкретных задач, и правильное обоснование принятых решений
- оценка «хорошо (7)» выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач некоторые неточности;
- оценка «хорошо (6)» выставляется студенту, если он знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач некоторые неточности;
- оценка «хорошо (5)» выставляется студенту, если он знает материал, и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач некоторые неточности;

- оценка «удовлетворительно (4)» выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, недостаточно правильные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, но при этом он владеет основными разделами учебной программы, необходимыми для дальнейшего обучения и может применять полученные знания по образцу в стандартной ситуации;
- оценка «удовлетворительно (3)» выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, недостаточно правильные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, но при этом он владеет фрагментарно основными разделами учебной программы, необходимыми для дальнейшего обучения и может применять полученные знания по образцу в стандартной ситуации;
- оценка «неудовлетворительно (2)» выставляется студенту, который не знает большей части основного содержания учебной программы дисциплины, допускает грубые ошибки в формулировках основных понятий дисциплины и не умеет использовать полученные знания при решении типовых практических задач
- оценка «неудовлетворительно (1)» выставляется студенту, который не знает формулировок основных понятий дисциплины

5. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности

Во время проведения экзамена обучающиеся могут пользоваться программой дисциплины.