**Санкт-Петербургский государственный университет**

**Р А Б О Ч А Я П Р О Г Р А М М А**

**УЧЕБНОЙ ДИСЦИПЛИНЫ**

Введение в архитектуру программ

Introduction to Program Architecture

**Язык(и) обучения**

русский

Трудоемкость в зачетных единицах: 3

Регистрационный номер рабочей программы: 002247

**Раздел 1. Характеристики учебных занятий**

**1.1. Цели и задачи учебных занятий**

Дисциплина «Введение в программную архитектуру» входит в перечень элективных курсов по специализации «Технологии программного обеспечения», формирующих продвинутую подготовку специалиста в области информатики.

Целью дисциплины является обучение организации процессов и методов:

* фиксации и сбора требований к программной системе, а также разработки, документирования и тестирования архитектурных проектов программных систем;
* принципам использования стандартных архитектурных шаблонов проектирования;
* подготовка к восприятию других дисциплин в области информатики.

Основной задачей курса является изучение архитектурных концепций, базовых моделей и технологий создания программной архитектуры, а также формирование у обучающихся современных представлений об основных этапах процесса проектирования и разработки крупных программных систем.

Основным методологическим принципом построения программы курса, равно как и всей концепции обучения в целом, является принцип поэтапного системного накопления знаний и формирования необходимых компетенций по модели: от простого и/или знакомого — к сложному и/или незнакомому, а основной методологической стратегией прохождения отдельных разделов программы является ступенчатость и цикличность, предусматривающие постепенный возврат к ранее усвоенному материалу на более высоком концептуальном уровне.

Главный принцип, который лежит в основе данной программы, — это следование концепции Европейского уровня работы с формализацией формулировок в области прикладной информатики и тем образовательным стандартам, которые обозначены этим документом в рамках приобретения компетенций, которые включают практические и теоретические компоненты.

По окончании обучения слушатели должны знать содержание дисциплины «Введение в архитектуру программ» и овладеть методами проектирования программных систем с использованием различных архитектурных концепций, соответствующих предъявляемым к системам требованиям, получить представления о способах анализа и оценки архитектурных проектов, критериях выбора проектных решений, наиболее адекватных требованиям к системам.

**1.2. Требования подготовленности обучающегося к освоению содержания учебных занятий (пререквизиты)**

Программа дисциплины рассчитана на обучающихся 3–го курса. Максимальная эффективность программы будет обеспечена при следующем условии: обучающийся владеет базовыми математическими понятиями и базовыми понятиями в области информатики, изученными на 1 – 2 курсах.

**1.3. Перечень результатов обучения (learning outcomes)**

* знать содержание дисциплины "Введение в программную архитектуру" и иметь достаточно полное представление о возможностях применения ее разделов в различных прикладных областях науки и техники;
* уметь применять полученные знания на практике.

Дисциплина способствует формированию следующих компетенций:

* ОПК-1 – способен применять фундаментальные знания, полученные в области математических и (или) естественных наук, и использовать их в профессиональной деятельности;
* ОПК-3 – способен применять современные информационные технологии, в том числе отечественные, при создании программных продуктов и программных комплексов различного назначения;
* ОПК-4 – способен участвовать в разработке технической документации программных продуктов и программных комплексов;
* ОПК-5 – способен инсталлировать и сопровождать программное обеспечение для информационных систем и баз данных, в том числе отечественного производства;
* ПКА-1 – способен демонстрировать базовые знания математических и естественных наук, программирования и информационных технологий;
* ПКП-1 – способность проводить под научным руководством исследование на основе существующих методов в конкретной области профессиональной деятельности;
* ПКП-2 – способен решать задачи в области развития науки, техники и технологии с учетом нормативного правового регулирования в сфере интеллектуальной собственности;
* ПКП-4 – способен применять современные информационные технологии при проектировании, реализации, оценке качества и анализа эффективности программного обеспечения для решения задач в различных предметных областях;
* ПКП-5 – способен использовать основные методы и средства автоматизации проектирования, реализации, испытаний и оценки качества при создании конкурентоспособного программного продукта и программных комплексов, а также способен использовать методы и средства автоматизации, связанные с сопровождением, администрированием и модернизацией программных продуктов и программных комплексов;
* ПКП-6 – способен использовать знания направлений развития компьютеров с традиционной (нетрадиционной) архитектурой; современных системных программных средств: операционных систем, операционных и сетевых оболочек, сервисных программ; тенденции развития функций и архитектур проблемно-ориентированных программных систем и комплексов в профессиональной деятельности;
* ПКП-8 – способен использовать современные методы разработки и реализации конкретных алгоритмов математических моделей на базе языков программирования и пакетов прикладных программ моделирования;
* УКБ-3 – способен понимать сущность и значение информации в развитии общества, использовать основные методы получения и работы с информацией с учетом современных технологий цифровой экономики и информационной безопасности.

**1.4. Перечень и объём активных и интерактивных форм учебных занятий**

Занятия в активных и интерактивных формах объёмом 4 ак. часа: лекции, предполагающие активную дискуссию с преподавателем.

**Раздел 2. Организация, структура и содержание учебных занятий**

**2.1. Организация учебных занятий**

**2.1.1 Основной курс**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Трудоёмкость, объёмы учебной работы и наполняемость групп обучающихся | | | | | | | | | | | | | | | | | | |
| Код модуля в составе дисциплины,  практики и т.п. | Контактная работа обучающихся с преподавателем | | | | | | | | | | | | Самостоятельная работа | | | | Объём активных и интерактивных  форм учебных занятий | Трудоёмкость |
| лекции | семинары | консультации | практические  занятия | лабораторные работы | контрольные работы | коллоквиумы | текущий контроль | промежуточная  аттестация | итоговая аттестация | под руководством преподавателя | в присутствии  преподавателя | сам. раб. с использованием  методических материалов | текущий контроль (сам.раб.) | промежуточная аттестация (сам.раб.) | итоговая аттестация  (сам.раб.) |
| ОСНОВНАЯ ТРАЕКТОРИЯ | | | | | | | | | | | | | | | | | | |
| Форма обучения: очная | | | | | | | | | | | | | | | | | | |
| Семестр 6 | 30 |  | 2 |  |  |  |  |  | 2 |  |  |  | 46 |  | 28 |  | 4 | 3 |
|  | 2-100 |  | 2-100 |  |  |  |  |  | 2-100 |  |  |  | 1-1 |  | 1-1 |  |  |  |
| ИТОГО | 30 |  | 2 |  |  |  |  |  | 2 |  |  |  | 46 |  | 28 |  |  | 3 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Виды, формы и сроки текущего контроля успеваемости и промежуточной аттестации | | | | | | |
| Код модуля в составе дисциплины, практики и т.п. | Формы текущего контроля успеваемости | | Виды промежуточной аттестации | | Виды итоговой аттестации  (только для программ итоговой аттестации и дополнительных образовательных программ) | |
| Формы | Сроки | Виды | Сроки | Виды | Сроки |
| ОСНОВНАЯ ТРАЕКТОРИЯ | | | | | | |
| Форма обучения: очная | | | | | | |
| Семестр 6 |  |  | экзамен, устно, традиционная форма | по графику промежуточной аттестации |  |  |

**2.2. Структура и содержание учебных занятий**

Курс обучения состоит из двух частей:

Часть 1. Основы разработки архитектуры программных систем (разделы I - III) – 32 часа

Часть 2. Описание, анализ и типовые образцы программной архитектуры (разделы IV – VI) – 32 часа.

**Лекции**

1. **Введение**

Назначение архитектуры.

Круг заинтересованных лиц, влияющих на архитектуру.

Архитектурное представление моделей системы.

(4 часа)

1. **Унифицированный процесс разработки ПО (RUP)**

Итеративный и инкрементный подход к разработке программного обеспечения (RUP). Основные артефакты и участники.

Стадия определения требований. Модель предметной области. Нахождение, детализация и расстановка по приоритету вариантов использования. Создание прототипа интерфейса пользователя.

Стадия анализа. Цели и роль анализа в жизненном цикле программы. Проведение анализа архитектуры, вариантов использования, классов и пакетов.

Стадия проектирования архитектуры. Реализация вариантов использования. Описание архитектуры с помощью подсистем, классов и интерфейсов. Модель развертывания.

Стадия реализации. Представление модели реализации. Компоненты, подсистемы и интерфейсы. План сборки.

Стадия тестирования. Планирование, разработка и реализация тестов. Процедура и оценка результатов тестирования.

(18 часов)

1. **Архитектурные стили и реализация атрибутов качества**

Архитектурные и программные структуры системы.

Атрибуты качества системы.

Тактики реализации: готовности, модифицируемости, производительности, безопасности, контролепригодности, практичности.

(10 часов)

1. **Документирование и реконструкция программной архитектуры.**

Представления архитектуры.

Документирование поведения и интерфейсов.

Перекрестная документация.

Извлечение информации из исходного кода.

Объединение представлений.

Создание и работа с базой данных реконструируемой системы.

(6 часов)

1. **Методы анализа и оценки программной архитектуры**

Анализ компромиссных архитектурных решений: презентация архитектуры, генерация дерева полезности атрибутов качества, двухэтапный анализ архитектурных методик.

Анализ стоимости и эффективности принятия архитектурных решений: уточнение сценариев, установление полезности, разработка для сценариев архитектурных стратегий, отбор стратегий реализации с учетом коэффициента ROI и других ограничений.

(6 часов)

1. **Типовые архитектурные решения разработки корпоративных приложений**

Особенности корпоративных приложений. Модель слоев системы.

Представление бизнес-логики: сценарий-транзакции, модель предметной области, модуль таблицы, слой служб.

Представление данных в Web: модель-представление-контроллер, контроллер страниц, контроллер запросов, представление по шаблону, представление с преобразованием, двухэтапное представление, контроллер приложения.

Сохранение сеанса на стороне клиента, сервера и в базе данных.

Распределенная обработка данных: интерфейс удаленного доступа, объект переноса данных.

Автономный параллелизм: оптимистическая и пессимистическая блокировки, блокировка с низкой степенью детализации, неявная блокировка.

Решения для источников данных: шлюз таблицы данных, шлюз записи данных, активная запись, преобразователь данных.

Объектно-реляционные решения для моделирования поведения: единица работы, коллекция объектов, загрузка по требованию.

Объектно-реляционные решения для моделирования структуры: поле идентификации, отображение внешних ключей, отображение через таблицу ассоциаций, отображение зависимых объектов, внедренное значение, сериализованный крупный объект, наследование с одной таблицей и таблицами для каждого класса, преобразователи наследования.

Объектно-реляционное отображение с использованием метаданных: отображение метаданных, объект запроса, хранилище.

(20 часов)

**Цель данного этапа обучения –**. **Знания и умения по завершении профессионально-ориентированного модуля**

К концу данного этапа обучения обучающийся должен:

1. знать основные концепции программной архитектуры;
2. уметь применять их на практике.

**Темы для изучения и обсуждения**

1. Определение и роль архитектуры в создании программных систем. Обязанности архитектора.
2. Описание архитектуры на основе моделей: вариантов использования, проектирования, реализации, развертывания.
3. Суть итеративной и инкрементной разработки. Ее основные преимущества. Критерии завершения каждой фазы проекта.
4. Процесс отбора требований к системе. Разработка модели предметной области и бизнес-модели.
5. Артефакты и сотрудники процесса определения требований. Создание прототипа интерфейса пользователя.
6. Нахождение актантов, детализация и расстановка приоритетов вариантов использования. Структурирование модели вариантов использования.
7. Фаза анализа: роль в жизненном цикле программы, артефакты, сотрудники, деятельность.
8. Сотрудники и артефакты, вовлеченные в процесс проектирования.
9. Проектирование архитектуры и вариантов использования.
10. Проектирование классов и подсистем.
11. Стадия реализация. Артефакты, сотрудники, рабочий процесс.
12. Этап тестирования: артефакты, сотрудники, деятельность.
13. Архитектурные структуры: модуль, компонент и соединитель, распределение.
14. Тактики реализации готовности.
15. Тактики реализации модифицируемости.
16. Тактики реализации производительности.
17. Тактики реализации безопасности.
18. Тактики реализации контролепригодности.
19. Тактики реализации практичности.
20. Варианты применения архитектурной документации
21. Представления. Выбор значимых представлений. Перекрестная документация.
22. Документирование представления. Документирование поведения и интерфейсов.
23. Операции в ходе реконструкции программной архитектуры. Извлечение информации и создание базы данных.
24. Объединение представлений и процесс реконструкции.
25. Анализ компромиссных архитектурных решений: презентация архитектуры, генерация дерева полезности атрибутов качества, двухэтапный анализ архитектурных методик.
26. Анализ стоимости и эффективности принятия архитектурных решений: уточнение сценариев, установление полезности, разработка для сценариев архитектурных стратегий, отбор стратегий реализации с учетом коэффициента ROI и других ограничений.
27. Особенности корпоративных приложений. Модель слоев системы.
28. Представление бизнес-логики: сценарий-транзакции, модель предметной области.
29. Представление бизнес-логики: модуль таблицы, слой служб.
30. Представление данных в Web: модель-представление-контроллер, контроллер страниц, контроллер запросов.
31. Представление данных в Web: представление по шаблону, представление с преобразованием, двухэтапное представление, контроллер приложения.
32. Сохранение сеанса на стороне клиента, сервера и в базе данных.
33. Распределенная обработка данных: интерфейс удаленного доступа, объект переноса данных.
34. Автономный параллелизм: оптимистическая блокировка, блокировка с низкой степенью детализации, неявная блокировка.
35. Автономный параллелизм: пессимистическая блокировка.
36. Решения для источников данных: шлюз таблицы данных, шлюз записи данных.
37. Решения для источников данных: активная запись, преобразователь данных.
38. Объектно-реляционные решения для моделирования поведения: единица работы, коллекция объектов.
39. Объектно-реляционные решения для моделирования поведения: загрузка по требованию.
40. Объектно-реляционные решения для моделирования структуры: поле идентификации, отображение внешних ключей.
41. Объектно-реляционные решения для моделирования структуры: отображение через таблицу ассоциаций, отображение зависимых объектов.
42. Объектно-реляционные решения для моделирования структуры: внедренное значение, сериализованный крупный объект.
43. Объектно-реляционные решения для моделирования структуры: наследование с одной таблицей и таблицами для каждого класса, преобразователи наследования.
44. Объектно-реляционное отображение с использованием метаданных: отображение метаданных, объект запроса, хранилище.

**Раздел 3. Обеспечение учебных занятий**

**3.1. Методическое обеспечение**

**3.1.1 Методические указания по освоению дисциплины**

Для освоения дисциплины обучающиеся должны посещать лекции и практические занятия, выполнять задания преподавателей.

**3.1.2 Методическое обеспечение самостоятельной работы**

При самостоятельном изучении теоретического материала и выполнении практических заданий целесообразно использовать рекомендованную основную и дополнительную литературу.

**3.1.3 Методика проведения текущего контроля успеваемости и промежуточной аттестации и критерии оценивания**

Промежуточная аттестация предполагает завершение разработки приложения, требования к которому обучающийся получает в начале семестра. В случае, если к моменту проведения зачета обучающийся не предоставил работающий прототип, на зачете ему предлагается список экзаменационных вопросов. Требования к приложению носят индивидуальный характер, однако предполагают реализацию его архитектуры по стандартам UML 2.0. Виды приложений, их размер и функциональность определяются для каждого обучающегося индивидуально. В том случае, когда преподаватель ожидает, что реализация проекта займет больше времени, чем обучающемуся выделяется для самостоятельной работы, поощряется их объединение в команды для совместной работы.

Зачет проводится в устной форме. Билет состоит из двух вопросов. Время подготовки ответа на вопросы билета составляет 45 минут.

Использование конспектов и учебников, а также электронных устройств хранения, обработки или передачи информации при подготовке и ответе на вопросы экзамена категорически запрещено. В случае обнаружения факта использования недозволенных материалов (устройств) составляется акт и экзаменуемый удаляется с зачета. После ответа на вопросы билета преподаватель задает несколько дополнительных вопросов, на основании оценки ответов на которые итоговая оценка по предмету может быть повышена или понижена.

Критерии выставления оценок:

* Оценка «отлично» (ECTS A) ставится за демонстрацию готового приложения, требования к которому были согласованы с преподавателем в начале семестра. Во время демонстрации необходимо объяснить архитектуру приложения и успешно представить требуемые артефакты в соответствии с каждой темой курса. В случае устного зачета оценка «отлично» (ECTS A) ставится за полностью раскрытый теоретический материал и правильные ответы на дополнительные вопросы преподавателя. В болонской шкале оценка может быть скорректирована в ту или иную сторону с учетом малозначительных погрешностей изложения или, напротив, углубленного изложения материала.
* Оценка «хорошо» ставится за изложенный теоретический материал билета (возможно с помощью наводящих подсказок преподавателя). Оценка ECTS B ставится в случае неточностей только в одном ответе. Оценка ECTS C – при наличии неточностей менее, чем в трех ответах.
* Оценка «удовлетворительно» (ECTS D и E). Допущены существенные ошибки, но обучающийся обладает необходимыми знаниями для их устранения под руководством преподавателя. Знание учебного материала в объеме, необходимом для дальнейшей учебы и профессиональной деятельности (основные определения, знание требуемой функциональности). Обучающийся знаком с основной литературой, рекомендованной программой.
* Оценка «неудовлетворительно» (ECTS F) выставляется, если не выполняются условия для получения оценок «отлично», «хорошо» и «удовлетворительно».

**3.1.4 Методические материалы для проведения текущего контроля успеваемости и промежуточной аттестации (контрольно-измерительные материалы, оценочные средства)**

*Примерный перечень вопросов к экзамену:*

1. Определение и роль архитектуры в создании программных систем. Обязанности архитектора.
2. Описание архитектуры на основе моделей: вариантов использования, проектирования, реализации, развертывания.
3. Суть итеративной и инкрементной разработки. Ее основные преимущества. Критерии завершения каждой фазы проекта.
4. Процесс отбора требований к системе. Разработка модели предметной области и бизнес-модели.
5. Артефакты и сотрудники процесса определения требований. Создание прототипа интерфейса пользователя.
6. Нахождение актантов, детализация и расстановка приоритетов вариантов использования. Структурирование модели вариантов использования.
7. Фаза анализа: роль в жизненном цикле программы, артефакты, сотрудники, деятельность.
8. Сотрудники и артефакты, вовлеченные в процесс проектирования.
9. Проектирование архитектуры и вариантов использования.
10. Проектирование классов и подсистем.
11. Стадия реализация. Артефакты, сотрудники, рабочий процесс.
12. Этап тестирования: артефакты, сотрудники, деятельность.
13. Архитектурные структуры: модуль, компонент и соединитель, распределение.
14. Тактики реализации готовности.
15. Тактики реализации модифицируемости.
16. Тактики реализации производительности.
17. Тактики реализации безопасности.
18. Тактики реализации контролепригодности.
19. Тактики реализации практичности.
20. Варианты применения архитектурной документации
21. Представления. Выбор значимых представлений. Перекрестная документация.
22. Документирование представления. Документирование поведения и интерфейсов.
23. Операции в ходе реконструкции программной архитектуры. Извлечение информации и создание базы данных.
24. Объединение представлений и процесс реконструкции.
25. Анализ компромиссных архитектурных решений: презентация архитектуры, генерация дерева полезности атрибутов качества, двухэтапный анализ архитектурных методик.
26. Анализ стоимости и эффективности принятия архитектурных решений: уточнение сценариев, установление полезности, разработка для сценариев архитектурных стратегий, отбор стратегий реализации с учетом коэффициента ROI и других ограничений.
27. Особенности корпоративных приложений. Модель слоев системы.
28. Представление бизнес-логики: сценарий-транзакции, модель предметной области.
29. Представление бизнес-логики: модуль таблицы, слой служб.
30. Представление данных в Web: модель-представление-контроллер, контроллер страниц, контроллер запросов.
31. Представление данных в Web: представление по шаблону, представление с преобразованием, двухэтапное представление, контроллер приложения.
32. Сохранение сеанса на стороне клиента, сервера и в базе данных.
33. Распределенная обработка данных: интерфейс удаленного доступа, объект переноса данных.
34. Автономный параллелизм: оптимистическая блокировка, блокировка с низкой степенью детализации, неявная блокировка.
35. Автономный параллелизм: пессимистическая блокировка.
36. Решения для источников данных: шлюз таблицы данных, шлюз записи данных.
37. Решения для источников данных: активная запись, преобразователь данных.
38. Объектно-реляционные решения для моделирования поведения: единица работы, коллекция объектов.
39. Объектно-реляционные решения для моделирования поведения: загрузка по требованию.
40. Объектно-реляционные решения для моделирования структуры: поле идентификации, отображение внешних ключей.
41. Объектно-реляционные решения для моделирования структуры: отображение через таблицу ассоциаций, отображение зависимых объектов.
42. Объектно-реляционные решения для моделирования структуры: внедренное значение, сериализованный крупный объект.
43. Объектно-реляционные решения для моделирования структуры: наследование с одной таблицей и таблицами для каждого класса, преобразователи наследования.
44. Объектно-реляционное отображение с использованием метаданных: отображение метаданных, объект запроса, хранилище.

**3.1.5 Методические материалы для оценки обучающимися содержания и качества учебного процесса**

Для оценки обучающимися содержания и качества учебного процесса применяется анкетирование в соответствии с методикой и графиком, утвержденными в установленном порядке.

**3.2. Кадровое обеспечение**

**3.2.1 Образование и (или) квалификация штатных преподавателей и иных лиц, допущенных к проведению учебных занятий**

К преподаванию дисциплины могут быть допущены преподаватели, имеющие диплом о высшем образовании по соответствующему направлению.

**3.2.2 Обеспечение учебно-вспомогательным и (или) иным персоналом**

Специальных требований нет.

**3.3. Материально-техническое обеспечение**

**3.3.1 Характеристики аудиторий (помещений, мест) для проведения занятий**

В аудиториях, где проводятся занятия, необходимо наличие доски.

**3.3.2 Характеристики аудиторного оборудования, в том числе неспециализированного компьютерного оборудования и программного обеспечения общего пользования**

Специальных требований нет.

**3.3.3 Характеристики специализированного оборудования**

Специальных требований нет.

**3.3.4 Характеристики специализированного программного обеспечения**

Специальных требований нет.

**3.3.5 Перечень и объёмы требуемых расходных материалов**

Расходные материалы не требуются.

**3.4. Информационное обеспечение**

**3.4.1 Список обязательной литературы**

1. Якобсон А., Буч Г., Рамбо Дж. Унифицированный процесс разработки программного обеспечения. – Спб.: Питер, 2002 г.
2. Басс Л., Клементс П., Кацман Р. Архитектура программного обеспечения на практике. 2-е издание. – Спб.: Питер, 2006 г.
3. Фаулер М. Шаблоны корпоративных приложений. – М. Издательский дом “Вильямс”, 2010 г.

**3.4.2 Список дополнительной литературы**

Не предусмотрено.

**3.4.3 Перечень иных информационных источников**

Ресурсы сети Интернет.

**Раздел 4. Разработчики программы**

Григорьев Д.А., к.ф.-м. н., старший преподаватель.