ИНСТИТУТ ИНТЕЛЛЕКТУАЛЬНЫХ КИБЕРНЕТИЧЕСКИХ СИСТЕМ

КАФЕДРА КИБЕРНЕТИКИ

ОДОБРЕНО  
  
протокол № 18 / 03   
  
от « 31 » мая 2018 г.

**РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ**

ПРОЕКТИРОВАНИЕ И АРХИТЕКТУРА ПРОГРАММНЫХ СИСТЕМ (DESIGN AND ARCHITECTURE OF SOFTWARE SYSTEMS)

|  |  |
| --- | --- |
| Направление подготовки (специальность) | 09.03.04 Программная инженерия |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Семестр** | **Интерактив** | **Трудоемкость, кред.** | **Общий объем курса, час.** | **Лекции, час.** | **Практич. занятия, час.** | **Лаборат. работы, час.** | **СРС, час.** | **КСР, час.** | **Форма(ы) контроля, экз./зач./КР/КП** |
| 7 |  | 3 | 108 | 32 | 32 | 0 | 44 | 0 | З |
| ИТОГО | 0 | 3 | 108 | 32 | 32 | 0 | 44 | 0 |  |

Группа: Б18-504, Б18-514

АННОТАЦИЯ

Дисциплина призвана обеспечить освоение студентами базовых теоретических знаний и практических приемов, характерных для разработки прикладных программных систем. Дисциплина также обеспечивает выработку знаний и навыков, позволяющих выполнять проектирование архитектуры и конструирование программного обеспечения.

1. ЦЕЛИ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

Целью освоения дисциплины является достижение следующих результатов образования:

Знания:

на уровне представлений: Основные понятия из области разработки программных систем, применяемые метрики, методы и инструментальные средства. Проблематика разработки программных систем в части архитектурного проектирования и конструирования.

на уровне воспроизведения: свойства моделей жизненного цикла, архитектур программных систем, особенности конструирования и проектирования программных систем.

на уровне понимания: Подходы к выбору моделей жизненного цикла, архитектур программных систем, а также к конструированию и проектированию программных систем в различных предметных областях и с применением различных языков программирования.

Умения:

теоретические – Основные задачи конструирования и проектирования программных систем и методы их решения. Методы ревизии программных продуктов. Подходы к документированию программных систем.

практические: - умение выбирать модель жизненного цикла, методологию разработки и архитектуру программной системы в соответствии со спецификой программной среды и предметной области. Выполнять проектирование, ревизию, разрабатывать документацию к программным системам.

навыки: - применять методы программной инженерии для решения задач конструирования и проектирования программных систем, с учетом специфики применяемых инструментальных средств и разрабатываемых предметных областей.

2. МЕСТО УЧЕБНОЙ ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП ВО

Дисциплина не требует специальной начальной подготовки, выходящей за рамки базового курса объектно-ориентированного программирования.

Дисциплина способствует формированию теоретических знаний и практических навыков в области в области разработки программных систем, позволяющих творчески применять их для решения задач архитектурного проектирования и конструирования программного обеспечения и обработки информации в профессиональной деятельности и научной работе.

3. КОМПЕТЕНЦИИ СТУДЕНТА, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ/ОЖИДАЕМЫЕ РЕЗУЛЬТАТЫ ОБРАЗОВАНИЯ И КОМПЕТЕНЦИИ СТУДЕНТА ПО ЗАВЕРШЕНИИ ОСВОЕНИЯ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ

ОПК-1 – Способен применять естественнонаучные и общеинженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности

ОПК-2 – Способен использовать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности

ОПК-3 – Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности

ОПК-4 – Способен участвовать в разработке стандартов, норм и правил, а также технической документации, связанной с профессиональной деятельностью

ПК-10 – владением методами контроля проекта и готовностью осуществлять контроль версий

ПК-11 – способность к формализации в своей предметной области с учетом ограничений используемых методов исследования

ПК-12 – готовность к использованию методов и инструментальных средств исследования объектов профессиональной деятельности

ПК-2 – владением навыками использования операционных систем, сетевых технологий, средств разработки программного интерфейса, применения языков и методов формальных спецификаций, систем управления базами данных

УК-1 – Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач

УК-3 – Способен осуществлять социальное взаимодействие и реализовывать свою роль в команде

УК-6 – Способен управлять своим временем, выстраивать и реализовывать траекторию саморазвития на основе принципов образования в течение всей жизни

4. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **№ п.п** | **Наименование раздела учебной дисциплины** | **Недели** | **Лекции, час.** | **Практ. занятия / семинары, час.** | **Лабораторные работы, час.** | **Обязат. текущий контроль (форма\*, неделя)** | **Аттестация раздела (форма\*, неделя)** | **Максимальный балл за раздел\*\*** |
|  | *7 Семестр* |  |  |  |  |  |  |  |
| 1 | Жизненный цикл и архитектуры программных систем | 1-6 | 12 | 12 | 0 | КИ-6,БДЗ-5 | КИ-6 | 20 |
| 2 | Проектирование программных систем | 7-13 | 14 | 14 | 0 | БДЗ-10,КИ-14 | КИ-14 | 20 |
| 3 | Конструирование программных систем | 14-16 | 6 | 6 | 0 | БДЗ-16,КР-16,КИ-16 | КИ-16 | 20 |
|  | *Итого за 7 Семестр* |  | 32 | 32 | 0 |  |  | 60 |
|  | **Контрольные мероприятия за 7 Семестр** |  |  |  |  |  | З | 40 |

\* – сокращенное наименование формы контроля

\*\* – сумма максимальных баллов должна быть равна 100 за семестр, включая зачет и (или) экзамен

Сокращение наименований форм текущего контроля и аттестации разделов:

|  |  |
| --- | --- |
| **Обозначение** | **Полное наименование** |
| БДЗ | Большое домашнее задание |
| КИ | Контроль по итогам |
| З | Зачет |

КАЛЕНДАРНЫЙ ПЛАН

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Недели** | **Темы занятий / Содержание** | **Лек., час.** | **Пр./сем., час.** | **Лаб., час.** |
|  | *7 Семестр* | 32 | 32 | 0 |
| **1-6** | **Жизненный цикл и архитектуры программных систем** | 12 | 12 | 0 |
| 1 | **Вводная лекция** Понятие программной системы. Процессы проектирования и конструирования программного обеспечения. Архитектуры программных систем. История развития. Цели и задачи учебной дисциплины. Применяемые методы и средства. Проблемы учебной дисциплины. Основные разделы курса. Структура учебной программы дисциплины (контрольные мероприятия в семестре). | Всего аудиторных часов | Всего аудиторных часов | Всего аудиторных часов |
| 2 | 2 |  |
| Онлайн | Онлайн | Онлайн |
|  |  |  |
| 2 | **Обзор жизненного цикла разработки программного обеспечения** Понятие жизненного цикла. Этапы жизненного цикла. Анализ и спецификация требований. Предварительное и детальное проектирование. Реализация и тестирование. Интеграция. Сопровождение. Экономика жизненного цикла | Всего аудиторных часов | Всего аудиторных часов | Всего аудиторных часов |
| 2 | 2 |  |
| Онлайн | Онлайн | Онлайн |
|  |  |  |
| 3 | **Модели жизненного цикла разработки программных систем** Общие принципы и структура. Модель Build-and-Fix. Водопадная модель. Быстрое прототипирование. Инкрементная модель. Модель синхронизации и стабилизации. Спиральная модель. Объектно-ориентированная модель. Преимущества и недостатки моделей. | Всего аудиторных часов | Всего аудиторных часов | Всего аудиторных часов |
| 2 | 2 |  |
| Онлайн | Онлайн | Онлайн |
|  |  |  |
| 4 | **Методологии разработки программных систем** Понятие методологии. Методология Microsoft Solution Framework. Подход Rational Unified Process. Процессы, роли, артефакты. Активности, потоки работ. Гибкие методологии (Agile, SCRUM, eXtreme Programming). Сходства и различия между методологией разработки и моделью жизненного цикла программной системы. Преимущества и недостатки методологий. | Всего аудиторных часов | Всего аудиторных часов | Всего аудиторных часов |
| 2 | 2 |  |
| Онлайн | Онлайн | Онлайн |
|  |  |  |
| 5 | **Открытые системы. Обзор архитектур программных систем** Определение открытой системы. Компоненты и интерфейсы. Понятие программной архитектуры. Основные типы программных архитектур. Архитектуры крупномасштабных программных систем. Понятие масштабируемости. Обеспечение масштабируемости за счет выбора адекватной программной архитектуры. Классификация программных архитектур. Архитектурное проектирование. Примеры типичных ошибок при проектировании архитектуры. | Всего аудиторных часов | Всего аудиторных часов | Всего аудиторных часов |
| 2 | 2 |  |
| Онлайн | Онлайн | Онлайн |
|  |  |  |
| 6 | **Архитектуры распределенных программных систем** Понятие распределенной программной системы. Архитектура «файл-сервер». Основные типы программных архитектур класса «клиент-сервер». Презентационная логика. Логика доступа к данным. Бизнес-логика. Двухуровневая и трехуровневая схема «клиент-сервер». Сервер баз данных. Интернет-архитектуры. Значение выбора архитектуры для проектирования программных систем. | Всего аудиторных часов | Всего аудиторных часов | Всего аудиторных часов |
| 2 | 2 |  |
| Онлайн | Онлайн | Онлайн |
|  |  |  |
| **7-13** | **Проектирование программных систем** | 14 | 14 | 0 |
| 7 | **CASE-cредства** Понятие метрики в программной инженерии. Основные метрики программного продукта. Инструментальные средства автоматизации разработки программных систем. Классификация и назначение CASE-средств. Связь CASE-средств с этапами проектирования программных систем. Требования к процессу проектирования и квалификации команды разработки. Условия эффективного применения CASE-средств. Преимущества и проблемы использования CASE-средств. | Всего аудиторных часов | Всего аудиторных часов | Всего аудиторных часов |
| 2 | 2 |  |
| Онлайн | Онлайн | Онлайн |
|  |  |  |
| 8 | **Введение в объектно-ориентированное проектирование** Сравнение структурного и объектно-ориентированного подходов. Особенности, экономика, артефакты. Объектно-ориентированная модель жизненного цикла программных систем. Преимущества и недостатки модели. Диаграммы. Особенности управления жизненным циклом объектно-ориентированных программных систем. Проблемы и приемы эффективной разработки. | Всего аудиторных часов | Всего аудиторных часов | Всего аудиторных часов |
| 2 | 2 |  |
| Онлайн | Онлайн | Онлайн |
|  |  |  |
| 9 | **Принципы объектно-ориентированного проектирования** Абстракция в математике и программировании. Математическая модель абстракции. Степень абстракции как средство оптимизации модели.  Инкапсуляция. Принцип «Разделяй и властвуй». Разделение интерфейса и реализации. Признаки инкапсуляции. Связь с модульной декомпозицией.  Наследование. Базовые и производные классы. Абстрактные классы и методы. Наследование реализации, поведения и свойств. Переопределение метода. Виды наследования. Интерфейсы. Проблема «хрупких» базовых классов.  Полиморфизм. Унификация и повторное использование кода. Связь c инкапсуляцией и наследованием. Расширения полиморфизма. Виды связывания. «Ленивые» вычисления.  Связь принципов с модульной декомпозицией. Примеры использования. | Всего аудиторных часов | Всего аудиторных часов | Всего аудиторных часов |
| 2 | 2 |  |
| Онлайн | Онлайн | Онлайн |
|  |  |  |
| 10 | **Язык моделирования UML** UML как язык и стандарт. История создания. Основные виды диаграмм (диаграммы классов, прецедентов, взаимодействия, деятельности, состояний). Применение UML для описания архитектуры программной системы. Моделирование отношений между классами. CASE-средства, поддерживающие UML-диаграммы. Особенности разработки UML-диаграмм в среде Microsoft Visual Studio .NET. | Всего аудиторных часов | Всего аудиторных часов | Всего аудиторных часов |
| 2 | 2 |  |
| Онлайн | Онлайн | Онлайн |
|  |  |  |
| 11 | **Стадии проектирования** Понятие объектно ориентированного проектирования. Последовательность действий при проектировании. Выявление первичных классов. Извлечение существительных. Уточнение отношений межу классами. Построение взаимосвязей классов. Сигнатуры классов. Поддержка проектирования диаграммами UML и CASE-средствами. Детализация диаграмм классов. Особенности и проблемы проектирования программных систем. | Всего аудиторных часов | Всего аудиторных часов | Всего аудиторных часов |
| 2 | 2 |  |
| Онлайн | Онлайн | Онлайн |
|  |  |  |
| 12 | **Многократное использование. Паттерны** Многократное использование артефактов программной системы. Виды артефактов. Экономика многократного использования. Понятие паттерна проектирования. Применение паттернов для многократного использования. Элементы паттернов проектирования. Предметно-ориентированные языки. Важность дисциплины при разработке программной системы. Поддержка паттернов проектирования диаграммами UML и CASE-средствами. | Всего аудиторных часов | Всего аудиторных часов | Всего аудиторных часов |
| 2 | 2 |  |
| Онлайн | Онлайн | Онлайн |
|  |  |  |
| 13 | **Пользовательские интерфейсы** Формы интерфейса пользователя. Значение независимости интерфейса пользователя от реализации. Применение паттернов для обеспечения независимости прикладной логики от интерфейса. Пути увеличения доли многократного использования программных компонентов. Эргономика интерфейсов пользователя. | Всего аудиторных часов | Всего аудиторных часов | Всего аудиторных часов |
| 2 | 2 |  |
| Онлайн | Онлайн | Онлайн |
|  |  |  |
| **14-16** | **Конструирование программных систем** | 6 | 6 | 0 |
| 14 | **Ревизия** Понятие ревизии. Важность раннего выявления ошибок проектирования. Взаимодействие фаз анализа, проектирования и реализации. Виды ревизии. Распределение ролей в группе контроля качества. Цель и методика ревизии. Объекты ревизии. Организация инспекции. Метрики и средства автоматизации ревизии. | Всего аудиторных часов | Всего аудиторных часов | Всего аудиторных часов |
| 2 | 2 |  |
| Онлайн | Онлайн | Онлайн |
|  |  |  |
| 15 | **Модульное тестирование** Классификация видов тестирования. Модульное, сборочное и приемочное тестирование. Важность тестирования качества и надежности программной системы. Подходы к модульному тестированию. Тестирование «от кода». Тестирование «от спецификаций». Методы «прозрачного ящика» и «черного ящика». Виды, метрики средства автоматизации модульного тестирования. | Всего аудиторных часов | Всего аудиторных часов | Всего аудиторных часов |
| 2 | 2 |  |
| Онлайн | Онлайн | Онлайн |
|  |  |  |
| 16 | **Реализация** Важность этапа реализации для обеспечения качества. Подходы к сборочному тестированию «от кода» и «от спецификаций». Формы и виды тестирования. Методы тестирования. Технология «чистого помещения». Применение классов эквивалентности. Использование CASE-средств. Восходящее, нисходящее и гибридное тестирование. Метрики и средства автоматизации сборочного тестирования. | Всего аудиторных часов | Всего аудиторных часов | Всего аудиторных часов |
| 2 | 2 |  |
| Онлайн | Онлайн | Онлайн |
|  |  |  |

Сокращенные наименования онлайн опций:

|  |  |
| --- | --- |
| **Обозначение** | **Полное наименование** |
| ЭК | Электронный курс |
| ПМ | Полнотекстовый материал |
| ПЛ | Полнотекстовые лекции |
| ВМ | Видео-материалы |
| АМ | Аудио-материалы |
| Прз | Презентации |
| Т | Тесты |
| ЭСМ | Электронные справочные материалы |
| ИС | Интерактивный сайт |

ТЕМЫ ПРАКТИЧЕСКИХ ЗАНЯТИЙ

|  |  |
| --- | --- |
| **Недели** | **Темы занятий / Содержание** |
|  | *7 Семестр* |
| 1 - 2 | **Обоснование выбора модели жизненного цикла** Обоснование выбора модели жизненного цикла |
| 3 | **Выбор методологии разработки программной системы** Выбор методологии разработки программной системы |
| 4 | **Согласование модели жизненного цикла с методологией разработки** Согласование модели жизненного цикла с методологией разработки |
| 5 | **Архитектурное проектирование программной системы** Архитектурное проектирование программной системы |
| 6 | **Согласование модели жизненного цикла, методологии разработки и архитектуры** Согласование модели жизненного цикла, методологии разработки и архитектуры |
| 7 | **Архитектурное проектирование при помощи CASE-средств** Архитектурное проектирование при помощи CASE-средств |
| 8 | **Первичное проектирование. Выявление первичных классов.**  Первичное проектирование. Выявление первичных классов. |
| 9 | **Разработка иерархии классов с детализацией абстрактных/полиморфных методов.** Разработка иерархии классов с детализацией абстрактных/полиморфных методов. |
| 10 | **Разработка диаграмм UML (прецеденты, классы).** Разработка диаграмм UML (прецеденты, классы). |
| 11 | **Детализация диаграмм классов** Детализация диаграмм классов |
| 12 | **Проектирование паттернов** Проектирование паттернов |
| 13 | **Проектирование интерфейсов пользователя.** Проектирование интерфейсов пользователя. |
| 14 | **Ревизия программного продукта.**  Ревизия программного продукта. |
| 15 - 16 | **Разработка плана тестирования, эксплуатации и сопровождения.** Разработка плана тестирования, эксплуатации и сопровождения. |
| 17 - 18 | **Разработка документации к программной системе.** Разработка документации к программной системе. |

5. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

1. Лекционные занятия:

a. комплект электронных презентаций/слайдов,

b. аудитория, оснащенная презентационной техникой (проектор, экран, компьютер/ноутбук)

2. Практические занятия:

a. компьютерный класс,

b. презентационная техника (проектор, экран, компьютер/ноутбук),

c. пакет программ Microsoft Office, включающий Microsoft Visio

d. стандартный пакет программ Microsoft Visual Studio.

6. ТРЕБОВАНИЯ К ФОНДУ ОЦЕНОЧНЫХ СРЕДСТВ В РАМКАХ РЕАЛИЗУЕМОЙ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

В качестве оценочного средства используется 100 бальная семестровая система, учитывающая посещаемость занятий, активность (выполнение домашних занятий), выполнение тематических домашних заданий по каждому разделу, контрольно-тестовая работа по каждому разделу. Каждый раздел проходит аттестацию.

ПЗ – посещаемость семинарских занятий (еженедельно) не менее 80% +2 балла

не менее 50% +1 балл

менее 50% 0 баллов

ТДЗ – выполнения тематического ДЗ (по каждому разделу)

Выполнено не менее 90% +10 баллов

Выполнено от 80-до 89% +8 балла

Выполнено от 70-до 79% +6 балла

Выполнено от 60-до 69% +4 балла

Выполнено от 40-до 59% +2 балл

Менее 39% 0 баллов

КТР - контрольно-тестовая работа (продолжительность – 1 а/час

(проводится в аудитории) Выполнено не менее 90% +8 баллов

Выполнено от 70-до 89% +6 баллов

Выполнено от 40-до 69% +4 балла

Менее 39% 0 баллов

АМ – аттестация раздела Раздел аттестуется, если набрано не менее 60%

Зачет (40 баллов). На зачет выносятся вопросы, относящиеся ко всем разделам. Зачет проводится в письменном виде по индивидуальному билету. Каждый билет содержит 10 заданий. Два из них - теоретические, которые выбираются из списка вопросов к зачету. Остальные задания связаны с проверкой теоретических и практических знаний по всем разделам дисциплины. Письменные ответы студента регистрируются на специальных бланках. Студент обязательно отмечает на этих бланках символом "+" те вопросы и задачи из билета, на которые даны полные о развернутые ответы. Отмечает символом "", если ответ не полон или решение задачи не выполнено до конца. Символом "-" отмечаются те вопросы (задачи), которые не нашли своего отражения в ответах. Каждый вопрос оценивается по следующей схеме:

• 4 балла (полный развернутый ответ на теоретический вопрос или полное и обоснованное решение практической задачи);

• 2 балла (ответ на теоретический вопрос не полон, имеются отдельные неточности в определениях и свойствах, получены частичные результаты решения практической задачи);

• 0 баллов (ответы на теоретический вопрос отсутствуют, обоснование оперирует ложными понятиями либо полностью отсутствует и т.п.).

7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

ОСНОВНАЯ ЛИТЕРАТУРА:

1. ЭИ Z99 Crisis Management for Software Development and Knowledge Transfer : , Cham: Springer International Publishing, 2016

2. ЭИ М 25 Программная инженерия : учебное пособие, Санкт-Петербург: Лань, 2018

ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА:

1. 004 С38 Верификация программного обеспечения : учебное пособие, С. В. Синицын, Н. Ю. Налютин, Москва: Интернет-Университет информационных технологий; Бином. Лаборатория знаний, 2008

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ:

Специальное программное обеспечение не требуется

LMS И ИНТЕРНЕТ-РЕСУРСЫ:

1. ИНТУИТ (http://intuit.ru)

8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

Специальное материально-техническое обеспечение не требуется

9. ##Definition not found: 'static\_section\_edu\_stud'##

Студенты, изучающие дисциплину "Конструирование программного обеспечения. Проектирование и архитектура программных систем", должны обратить внимание на широту и глубину знаний, требуемых для успешного освоения этой дисциплины. При этом на дисциплину отводится сравнительно небольшое количество часов аудиторных занятий. Отсюда следует, что результат успешного освоения дисциплины во многом определяется самостоятельной работой студентов. В связи с этим можно рекомендовать регулярный просмотр сайтов зарубежных и отечественных компаний, занимающихся разработкой программных систем. Подобные ссылки даются на лекциях и практических занятиях. Следует также участвовать в конференциях и семинарах, которые регулярно проводятся в Москве ведущими компаниями-разработчиками программных продуктов. Стоит обратить внимание на следующие сайты:

1) http://events.cnews.ru/events/

2) http://www.ibm.com/ru/events/

3) http://www.academy.it.ru/ru/about/actions/

4) http://www.techdays.ru/

5) https://msevents.microsoft.com/cui/default.aspx?culture=ru-ru

6) http:// h41110.www4.hp.com/ur/institute.html

и многие другие. Многие ведущие программные компании (IBM, MS, HP и др.) реализуют программы академической инициативы, которые позволяют студентам получать бесплатно программные продукты, проходить курсы обучения, участвовать в семинарах и конференциях.

Рабочая программа дисциплины составлена в соответствии с ОС НИЯУ МИФИ (ФГОС) и учебным планом основной образовательной программы (программ).

Автор(ы):

|  |  |  |
| --- | --- | --- |
|  | Зыков Сергей Викторович, к.т.н., доцент |  |