

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Курганский государственный университет»
(КГУ)

Кафедра «Программное обеспечение автоматизированных систем»



УТВЕРЖДАЮ:
Первый проректор
Т.Р. Змызгова
«05» сентября 2022 г.

Рабочая программа учебной дисциплины
ТЕХНОЛОГИИ И МЕТОДЫ ПРОГРАММИРОВАНИЯ

образовательной программы высшего образования –
программы специалитета

10.05.03 – Информационная безопасность автоматизированных систем

Направленность:

Безопасность открытых информационных систем

очная форма обучения


Курган 2022

Программа дисциплины «Технологии и методы программирования» составлена в соответствии с учебным планом программы специалитета 10.05.03 – Информационная безопасность автоматизированных систем (*Безопасность открытых информационных систем*), утвержденным 30 августа 2022 года.

Рабочая программа одобрена на заседании кафедры «Программное обеспечение автоматизированных систем» 02.09.2022 года, протокол № 2.

Рабочую программу разработал:

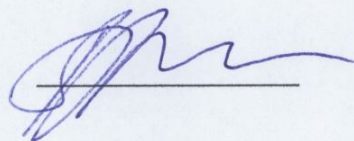
доцент кафедры ПОАС



В.К. Волк

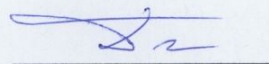
Согласовано:

Заведующий
кафедрой ПОАС



В.К. Волк

Заведующий
кафедрой БИАС



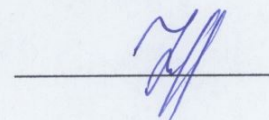
Д.И. Дик

Начальник
Управления
образовательной деятельности



И.В. Григоренко

Специалист
по учебно-методической работе
Учебно-методического отдела



Г.В. Казанкова

1. ОБЪЕМ ДИСЦИПЛИНЫ

Всего: 4 зачетных единицы (144 академических часа)

Виды учебной работы	Распределение трудоемкости по семестрам и видам учебных занятий, акад. часов	
	Всего	4-й семестр
Общая трудоемкость дисциплины	144	144
Аудиторные занятия, все-	64	64
Лекции	32	32
Лабораторные работы	32	32
Самостоятельная работа, всего:	80	80
Подготовка к экзамену	27	27
Другие виды самостоятельной работы	53	53
Виды промежуточной аттестации	Экзамен	Экзамен

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Дисциплина «Технологии и методы программирования» включена в модуль "Информатика и программирование" обязательной части образовательной программы, для ее освоения необходимы компетенции, сформированные при изучении дисциплин «Информатика» и «Основы программирования».

Результаты изучения дисциплины необходимы для освоения технологий проектирования информационных систем различного назначения и используются при освоении дисциплин "Безопасность систем баз данных", "Технология построения защищенных информационных приложений".

3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

Цель изучения дисциплины – ознакомление с технологиями и стандартами промышленной разработки программного обеспечения (далее - ПО) и получение практического опыта проектирования программных систем.

Задачами дисциплины являются изучение:

- понятий, методических основ и стандартов программной инженерии;
- основных моделей жизненного цикла ПО;
- промышленных технологий проектирования ПО;
- языка моделирования UML;
- стандартов документирования программных проектов;
- инструментальными CASE-средств.

Компетенции, формируемые в результате освоения дисциплины:

- способность создавать программы на языках общего назначения, применять методы и инструментальные средства программирования для

решения профессиональных задач, осуществлять обоснованный выбор инструментария программирования и способов организации программ (ОПК-7);

- разрабатывать компоненты систем защиты информации автоматизированных систем (ОПК-11);
- организовывать и проводить диагностику и тестирование систем защиты информации автоматизированных систем, проводить анализ уязвимостей систем защиты информации автоматизированных систем (ОПК-13)

В результате изучения дисциплины обучающийся

Должен знать:

- языки, системы и инструментальные средства, используемые в процессе проектирования программного обеспечения (для ОПК-7);
- стандарты и технологии промышленной разработки программного обеспечения и типовые проектные решения по обеспечению безопасности автоматизированных систем (для ОПК-11, ОПК-13);

Должен уметь:

- осваивать новые образцы программных, технических средств и информационных технологий (для ОПК-7);
- применять языки, системы и инструментальные средства проектирования программного обеспечения (для ОПК-7, ОПК-11);
- разрабатывать и анализировать проектные решения по обеспечению безопасности автоматизированных систем (для ПК-11);

Должен владеть практическими навыками применения инструментальных средств, используемых для документирования программных проектов (для ОПК-7);

4 СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

4.1 Учебно-тематический план

Разделы дисциплины		Часов контактной работы с преподавателем	
№	Наименование	Лекции	Лаб. работы
1	Технологии промышленной разработки ПО	30	30
Рубежный контроль №1		2	-
Рубежный контроль №2		-	2
Всего по дисциплине:		32	32

4.2 Курс лекций

Краткое содержание лекции	Часов контактной работы с преподавателем
<p><i>1. Предмет и базовые понятия программной инженерии.</i> Проблема увеличения сложности, стоимости и сроков разработки ПО. История развития технологий разработки программных продуктов. Понятие инженерной деятельности. Инженеры и программные инженеры. Понятие и ролевая модель команды программного проекта. Обзор профессиональных стандартов в IT-области: профессии IT-специалистов; функции и должностные обязанности; требования к квалификационным уровням и профессиональной компетентности.</p>	2
<p><i>2. Жизненный цикл ПО.</i> Понятие <i>жизненного цикла</i> (ЖЦ) промышленного изделия, как последовательности взаимосвязанных процессов, реализуемых на всех стадиях его существования. Проектирование как процесс преобразования информационных моделей создаваемого объекта. Декомпозиция, многоступенчатость и итерационность как базовые методы "борьбы со сложностью" проекта. Типовые стадии проекта. Проектная, программная и эксплуатационная документация. Обзор стандартов ЕСПД.</p>	4
<p><i>3. Процессы жизненного цикла программного продукта.</i> Обзор стандарта ISO/IEC 12207. Классификация и иерархия процессов ЖЦ ПО: основные, вспомогательные и организационные процессы. Участники процессов ЖЦ ПО. Основные процессы ЖЦ ПО: "заказ", "поставка", "разработка", "эксплуатация", "сопровождение". Типовая структура процесса "разработка": подготовка процесса; анализ требований к системе; проектирование системной архитектуры; анализ требований к программным модулям; проектирование программной архитектуры; техническое проектирование программных модулей; программирование, тестирование, сборка и квалификационные испытания программных модулей; сборка системы; квалификационные испытания системы; ввод в действие и обеспечение приемки.</p>	4

Краткое содержание лекции	Часов контактной работы с преподавателем
<p>4. <i>Модели жизненного цикла ПО.</i> Понятие модели ЖЦ ПО. Обзор типовых моделей ЖЦ ПО (каскадная, эволюционная и спиральная модели). Промышленные технологии создания ПО: технология <i>Rational Unified Process (RUP)</i>; технология <i>Microsoft Solution Framework (MSF)</i>, технология <i>eXtreme Programming (XP)</i>.</p>	4
<p>5. <i>Визуальное моделирование при анализе и проектировании программных систем.</i> Проектирование как процесс преобразования информационных моделей объекта. Задачи и базовые принципы проектирования сложных объектов (абстрагирование, декомпозиция, многоэтапность, итерационность). Визуализация при проектировании. Обзор систем графического моделирования. Структурные модели анализа бизнес-процессов: схема Захмана. Диаграммы структурного анализа систем: SADT - диаграммы функционального моделирования; ERD - диаграммы "Сущность – Связь"; DFD - диаграммы потоков данных. ООАП - объектно-ориентированный подход к анализу и проектированию программных систем. Язык графического моделирования UML: история стандартизации UML; структура и базовые понятия языка; модели и UML-диаграммы.</p>	4
<p><i>Рубежный контроль №1</i></p>	2
<p><i>UML-диаграммы вариантов использования (UseCase)</i> Use Case-модель как результат функциональной декомпозиции проектируемой системы на начальной стадии программного проекта. Назначение и область применения Use-Case-моделей. Компоненты модели и графическая нотация. Сценарии вариантов использования. Примеры UseCase-диаграмм.</p>	4
<p><i>UML-диаграммы классов и пакетов (Class- и Package-diagram)</i> Статические модели концептуального и логического уровней. «Пакеты» и «Классы» – два уровня структурной декомпозиции проектируемой системы. Компоненты моделей (пакеты, классы, интерфейсы, отношения), графическая нотация. Примеры диаграмм.</p>	4
<p><i>UML-диаграммы деятельности и диаграммы состояний (Activity-diagram и State Machine-diagram)</i> Назначение и область применения динамических UML-моделей. Модели логического уровня. Компоненты модели: простые и составные состояния; события и переходы; триггеры. Обозначения и графическая нотация. Примеры диаграмм.</p>	4
<p>Всего часов лекций по дисциплине:</p>	32

4.3 Лабораторный практикум

Номер лаб. работы	Наименования разделов и лабораторных работ	Часов контактной работы с преподавателем
1	<i>Подготовка к выполнению учебного программного проекта</i> Рассмотрение примера выполнения программного проекта. Выбор варианта темы проекта. Постановка задачи разработки.	6
2	<i>Разработка обобщенной UseCase-модели</i> <u>Стадия технического задания</u> : формирование терминологического словаря предметной области; разработка обобщенной диаграммы вариантов использования.	6
3	<i>Разработка диаграммы пакетов, UseCase-моделей и сценариев вариантов использования</i> <u>Стадия эскизного проекта</u> : структурная декомпозиция проектируемой системы (разработка диаграммы пакетов); функциональная декомпозиция подсистем (разработка локальных Use Case-диаграмм и сценариев вариантов использования)	6
4	<i>Разработка диаграммы классов</i> <u>Стадия технического проекта</u> – разработка статической модели проектируемой системы (компонента системы): структурная декомпозиция пакетов, разработка диаграмм классов.	6
5	<i>Разработка диаграммы состояний</i> <u>Стадия технического проекта</u> – разработка динамической модели поведения проектируемой системы (компонента системы): разработка диаграммы состояний (<i>StateMachine-diagram</i>).	6
Рубежный контроль №2		2
Всего часов лабораторных занятий по дисциплине		32

5. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ СТУДЕНТОВ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

5.1 Курс лекций

Лекционный курс базируется на пассивном методе обучения, реализующем традиционную объяснительно-иллюстративную образовательную технологию, в рамках которой студенты выступают в роли слушателей, воспринимающих учебный материал и участвующих в дискуссиях и экспресс-опросах.

Конспект лекций (краткий обзор рассматриваемых на лекциях вопросов) представлен в формате мультимедийных презентаций и включен в состав учебно-методического комплекса дисциплины, доступного студентам. Более детальное содержание лекционного материала представлено в соответствующих учебных пособиях, структура которых соответствует тематическому плану изучения дисциплины. Учебные пособия содержат перечни контрольных вопросов, ответы на которые должны быть получены студентами в процессе самостоятельной проработки материала соответствующей лекции.

5.2 Лабораторный практикум

Лабораторные работы объединены общей темой учебного программного проекта, в рамках которого последовательно разрабатываются соответствующие UML-диаграммы. Каждая лабораторная работа реализует одну из стадий программного проекта и требует освоения и применения соответствующих CASE-средств. Защита проектов проводится в форме собеседования по материалу представленного отчета и сделанного публичного доклада. В процессе защиты оценивается полнота и качество выполнения практических заданий каждым из участников команды проекта, грамотность использования инструментальных средств, качество оформления отчета.

5.3 Самостоятельная работа

Самостоятельная работа по освоению дисциплины включает изучение теоретических разделов дисциплины, подготовку к лекционным занятиям, выполнение лабораторных работ, подготовку к рубежному контролю и промежуточной аттестации.

Рекомендуемая трудоемкость самостоятельной работы представлена в таблице:

Виды самостоятельной работы	Рекомендуемая трудоемкость, акад. часов
<i>Изучение материала лекционного курса</i>	20
<i>Подготовка и выполнение лабораторных работ</i>	30
<i>Подготовка к рубежному контролю</i>	3
<i>Подготовка и проведение экзамена</i>	27
Всего:	80

6. ПРОЦЕДУРА ОЦЕНИВАНИЯ РЕЗУЛЬТАТОВ ОСВОЕНИЯ ДИСЦИПЛИНЫ

6.1 Состав и формы проведения контрольно-аттестационных мероприятий

Программой изучения дисциплины предусмотрены мероприятия текущего и рубежного контроля и промежуточная аттестация (в форме экзамена).

Текущий контроль проводится в форме защиты отчетов по выполненным лабораторным работам с демонстрацией полученных результатов. В процессе защиты оценивается своевременность и качество выполнения заданий, качество программного кода/проектных решений, а также эрудиция студента в вопросах, связанных с тематикой выполненной работы.

Рубежный контроль проводится в форме фронтального тестирования по основным разделам (темам) дисциплины. Оценивается количество правильных ответов на задания теста и соответственно начисляется балл. Студент, ответивший правильно менее, чем на 50% заданий теста, считается не прошедшим тестирование и обязан повторно пройти этот тест во время консультации по дисциплине.

Промежуточная аттестация по дисциплине в форме экзамена проводится в традиционной (устной) форме. Экзаменационный билет содержит 2 теоретических вопроса и одну задачу. На подготовку ответа студенту отводится один астрономический час. Оценивается качество ответа и решения задачи (максимально 10 баллов за ответ на один вопрос или задачу).

К экзамену допускаются студенты, выполнившие все лабораторные работы и успешно прошедшие процедуры рубежного контроля, предусмотренные в соответствующем семестре. Дополнительным критерием допуска к промежуточной аттестации является индивидуальная балльная оценка результатов работы в течение семестра (п. 6.3 рабочей программы).

6.2 Система балльно-рейтинговой оценки работы студентов

Оценивание результатов выполнения студентами плановых контрольных и аттестационных мероприятий по дисциплине производится в соответствии с Положением о балльно-рейтинговой системе контроля и оценки академической активности обучающихся ФГБОУ ВО «Курганский государственный университет».

Оценивание производится по 100-балльной (за семестр) шкале с последующим приведением итоговой 100-балльной рейтинговой оценки к традиционной четырех-балльной.

Рейтинговая оценка студента получается путем суммирования баллов, полученных студентом в течение семестра (максимум 70 баллов) и баллов, полученных им на промежуточной аттестации (максимум 30 баллов).

Минимальное количество баллов, которыми может быть оценен удовлетворительный ответ студента на экзамене, равно 11. Неудовлетворительный ответ оценивается в 0 баллов.

Максимальные балльные оценки по результатам проведения контрольных и аттестационных мероприятий приведены в следующих таблицах:

Виды контроля / аттестации	Содержание	Максимальная оценка	
		За единицу	Всего
Текущий контроль	Контроль посещения лекционных занятий	1	15
	Выполнение лабораторных работ (№2 - №5)	10	40
Рубежный контроль	№1. Стандарты и модели жизненного цикла ПО	5	5
	№2. Визуальное моделирование при проектировании ПО	10	10
Промежуточная аттестация (экзамен)		30	30
Максимальная итоговая оценка, баллов			100

Пересчет 100-балльной рейтинговой оценки студента в традиционную (4-балльную) оценку и в оценку ECTS (Общеввропейская система учета учебной работы) производится в соответствии с таблицей:

Рейтинговая оценка, баллов	Виды оценок промежуточной аттестации		
	Традиционная оценка		Оценка ECTS
91-100	5	Отлично	A
84-90	4	Очень хорошо	B
74-83		Хорошо	C
68-73	3	Удовлетворительно	D
61-67		Посредственно	E
31-60	2	Неудовлетворительно	Fx
0-30			F

6.3 Критерии допуска к промежуточной аттестации

Для допуска к промежуточной аттестации обучающийся должен выполнить все лабораторные работы и набрать по итогам текущего и рубежного контроля не менее 51 балла. В случае, если обучающимся набрана сумма менее 51 балла, он может набрать недостающее количество баллов, выполнив до конца последней (зачетной) недели семестра дополнительные задания, полученные им от преподавателя.

Обучающийся имеет право на получение итоговой оценки по дисциплине без проведения процедуры промежуточной аттестации, если он набрал не менее 61 балла по итогам текущего и рубежного контроля – в этом случае итоговая оценка определяется суммой баллов, набранных по итогам текущего и рубежного контроля. Обучающийся, имеющий право на получение итоговой оценки по результатам текущего и рубежного контроля, может повысить ее, пройдя процедуру промежуточной аттестации – в этом случае при получении на аттестационном испытании 0 баллов итоговая балльная оценка не снижается.

По усмотрению преподавателя, итоговая балльная оценка обучающегося может быть повышена за счет получения им дополнительных баллов за академическую активность (не более 30 дополнительных баллов).

Основанием для получения дополнительных баллов являются:

- выполнение дополнительных заданий по дисциплине;
- участие в течение семестра в учебной, научно-исследовательской, спортивной, культурно-творческой и общественной деятельности КГУ.

6.4 Фонд оценочных средств

6.4.1 Перечень оценочных средств

- 1) Балльно-рейтинговая система контроля о оценки академической активности студентов КГУ.
- 2) Задания для тестирования (2 рубежных контроля).
- 3) Вопросы для подготовки к экзамену по дисциплине.
- 4) Образцы отчетов по выполнению лабораторных работ.

Банк заданий для проведения мероприятий рубежных контроля и промежуточной аттестации по дисциплине, показатели, критерии, шкалы оценивания компетенций, а также методические материалы, определяющие процедуры оценивания образовательных результатов, приведены в учебно-методическом комплексе дисциплины.

6.4.2 Примерные варианты компонентов фонда оценочных средств

Вопросы для подготовки к экзамену:

1. Понятия "*программный продукт*" и "*программное обеспечение*" (software). "*Коробочные*" и "*заказные*" программные продукты. Концепции программной инженерии как промышленной технологии разработки и сопровождения ПО.
2. Единая система программной документации (ЕСПД). Требования к документированию программных проектах (стадии технического задания, эскизного, технического и рабочего проекта).
3. Понятие "*жизненного цикла ПО*". Типовая ролевая модель команды программного проекта. Основные функции *разработчика, тестировщика и инженера по качеству*.
4. Стандарт ISO/IEC 12207. Терминология. Классификация и структура процессов ЖЦ ПО.
5. Концепции модульного, структурного и объектно-ориентированного программирования.
6. Что такое CASE ? Приведите классификацию CASE-средств по различным классификационным признакам. Приведите примеры и рассмотрите функциональные возможности известных Вам CASE-средств уровня Low CASE.
7. Определите и дайте краткую характеристику основным *нефункциональным требованиям* к программному продукту.
8. Классы защищенности программных систем.
9. Способы повышения уровня защищенности компьютерных систем от преднамеренных внешних воздействий.

10. Понятие модели жизненного цикла ПО. Особенности каскадной, эволюционной и спиральной моделей.
11. Технология Rational Unified Process (RUP). Иерархическая структура модели ЖЦ RUP: фазы, итерации, рабочие и поддерживающие процессы, практики.
12. Технология Microsoft Solution Framework (MSF). Область применения и особенности. Основные фазы модели ЖЦ MSF.
13. Технология eXtreme Programming (XP). Базовые принципы методов "живой" разработки ПО. Основные фазы модели ЖЦ XP. Правила технологии XP

Тестовые задания для рубежного контроля №1:

1. Какая их типовых моделей жизненного цикла ПО может быть рекомендована к использованию при его разработке в условиях, когда требования заказчика могут изменяться в процессе проектирования ?
 - а) Каскадная модель
 - б) Эволюционная модель
 - в) Спиральная модель
 - г) Модель формальной разработки
 - д) Ни одна из перечисленных моделей
2. Какая из технологий разработки ПО базируется на следующих принципах:
 - люди, их общение более важны, чем процессы и инструменты;
 - работающая программа более важна, чем исчерпывающая документация;
 - сотрудничество с заказчиком более важно, чем обсуждение деталей контракта; отработка изменений более важна, чем следование планам.
 - а) Rational Unified Process (RUP)
 - б) Extreme programming (XP)
 - в) Microsoft solution framework (MSF)
 - г) Ни одна из перечисленных технологий

Тестовые задания для рубежного контроля №2:

1. При использовании какого подхода UML предоставляет максимум преимуществ?
 - а) Процедурное программирование
 - б) Объектно-ориентированное проектирование
 - в) Функциональное программирование
 - г) Программирование по контракту
 - д) Концептуальное проектирование
2. Выберите из списка истинные утверждения, касающиеся UML-моделей:
 - а) UML-модели являются XML-документами
 - б) UML имеет ограничения по природе моделируемой предметной области

- в) CASE-средства могут генерировать текстовые спецификации из UML-моделей
 - г) Создавая UML-модель, вы тем самым документируете систему
 - д) UML-модель жестко привязана к конкретной методологии разработки ПО.
3. Что специфицирует *отношение обобщения* между «актерами» на UML-диаграмме вариантов использования?
 4. Что специфицирует *отношение включения* между вариантами использования на UseCase-диаграмме вариантов использования?
 5. Что специфицирует *отношение обобщения* между классами на UML-диаграмме классов?
 6. Определите понятия «событие» и «переход», используемые при разработке State Machine-диаграмм.
 7. Определите понятия «состояние», «составное состояние», «параллельное состояние» и «последовательное состояние», используемые при разработке State Machine-диаграмм.
 8. Определите понятия «историческое состояние», «глубокое историческое состояние», используемые при разработке State Machine-диаграмм.

7 ОСНОВНАЯ И ДОПОЛНИТЕЛЬНАЯ УЧЕБНАЯ ЛИТЕРАТУРА

7.1. Основная литература

1. Буч Г. и др. Объектно-ориентированный анализ и проектирование с примерами приложений на C++, 3-е издание: пер. с англ. – Москва: Издательский дом «Вильямс», 2010 – 720 с.
2. Волк В.К. Практическое введение в программную инженерию: учебное пособие. – СПб.: Изд-во «Лань», 2019. – 100 с.: ил. – (Учебники для вузов. Специальная литература).

7.2. Дополнительная литература

3. Буч Г., Рамбо Дж., Джекобсон А. UML. Руководство пользователя. - ДМК-Пресс, Питер, 2004
4. Дик Д.И. Требования к оформлению текстовой документации курсовых и дипломных проектов (работ). Часть 1. Методические указания. – Курган : Изд-во Курганского гос. ун-та, 2008, –35 с.
5. Леоненков А. Самоучитель UML – 2-е издание.: – Санкт-Петербург: БХВ-Петербург.
6. Соммервилл И. Инженерия программного обеспечения – 6-е издание: пер. с англ. – Москва: Издательский дом «Вильямс», 2002. – 624 с.: ил

7.4 Информационно-справочные материалы

7. Автоматизированные системы. Защита от несанкционированного доступа к информации. Классификация автоматизированных систем и требования по защите информации. – Руководящий документ ФСТЭК. URL: <https://fstec.ru/component/attachments/download/296>

8. Классы защищенности для автоматизированных систем.
URL: <http://infoprotect.net/note/klassy-zashhity-informatsii#2>.
9. Единая система программной документации (ЕСПД). – Москва: Стандартинформ, 2005. URL: <http://www.swrit.ru/gost-esp.html>
 - ГОСТ 19.101-77. Виды программ и программных документов.
 - ГОСТ 19.102-77. Стадии разработки.
 - ГОСТ 19.103-77. Обозначения программ и программных документов.
 - ГОСТ 19.105-78. Общие требования к программным документам.
 - ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению.
 - ГОСТ 19.202-78. Спецификация. Требования к содержанию и оформлению.
 - ГОСТ 19.301-79 Программа и методика испытаний. Требования к содержанию и оформлению.
 - ГОСТ 19.401-78 Текст программы. Требования к содержанию и оформлению.
 - ГОСТ 19.402-78 Описание программы. Требования к содержанию и оформлению.
10. ГОСТ Р ИСО/МЭК 12207-99. Информационная технология. Процессы жизненного цикла программных средств.
11. StarUML™. The Open Source UML/MDA Platform. Руководство пользователя (электронный ресурс).
12. UML 2.5 Diagrams Overview. URL: <https://www.uml-diagrams.org/>.

8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Материально-техническое обеспечение дисциплины включает в себя учебные лаборатории и классы, оснащенные современными компьютерами (все – в стандартной комплектации для лабораторных занятий и самостоятельной работы), объединенными в локальную вычислительную сеть с выходом в Интернет; мультимедийное оборудование (переносной персональный компьютер, мультимедийный проектор, проекционный экран).

Программные средства обеспечения учебного процесса включают: операционные системы (Windows); инструментальные средства программирования и CASE-средства проектирования (StarUML™. The Open Source UML/MDA Platform); вспомогательные (программы презентационной графики; текстовые редакторы; графические редакторы).

9 ИСПОЛЬЗОВАНИЕ ДИСТАНЦИОННЫХ ОБРАЗОВАТЕЛЬНЫХ ТЕХНОЛОГИЙ

При использовании электронного обучения (ЭО) и дистанционных образовательных технологий (ДОТ) аудиторские занятия, а также текущий/рубежный контроль и промежуточная аттестация по дисциплине полностью или частично проводятся в режиме онлайн.

Решение об используемых ДОТ, системе оценивания достижений студентов и видах учебных занятий, проводимых в режиме онлайн, принимается кафедрой с учетом мнения ведущего преподавателя и доводится до сведения студентов, изучающих дисциплину.

Аннотация
рабочей программы учебной дисциплины
ТЕХНОЛОГИИ И МЕТОДЫ ПРОГРАММИРОВАНИЯ
образовательной программы высшего образования –
программы специалитета 10.05.03
Информационная безопасность автоматизированных систем
Направленность:
Безопасность открытых информационных систем
Форма обучения: **очная**

Трудоемкость дисциплины: 4 ЗЕ (144 академических часа)

Семестры: 4-й

Форма промежуточной аттестации: Экзамен,

Содержание дисциплины

Технологии промышленной разработки ПО.

Предмет и основные понятия программной инженерии.

Жизненный цикл ПО: базовые понятия. Модели жизненного цикла ПО.

Визуальное моделирование при анализе и проектировании программных систем.

Выявление и анализ функциональных требований. UML-диаграммы вариантов использования.

Разработка логической модели программной системы. UML-диаграммы классов.

Разработка динамических моделей программной системы.