


МИНОБРНАУКИ РОССИИ

Ярославский государственный университет им. П.Г. Демидова

Кафедра вычислительных и программных систем

УТВЕРЖДАЮ
Декан факультета ИВТ

 Д.Ю. Чалый

« 18 » мая 20 21 г.

Рабочая программа дисциплины
«Программная инженерия»

Направление подготовки
09.03.03 Прикладная информатика

Направленность (профиль)
«Информационные технологии в цифровой экономике»

Форма обучения
очная

Программа одобрена
на заседании кафедры
от «23» апреля 2021 года, протокол № 8

Программа одобрена НМК
факультета ИВТ
протокол № 7 от «17» мая 2021 года

Ярославль
2021

1. Цели освоения дисциплины

Дисциплина «Программная инженерия» обеспечивает приобретение знаний и умений в соответствии с государственным образовательным стандартом, содействует формированию навыков владения современными информационными технологиями и методами их разработки.

Цель дисциплины «Программная инженерия» – изучение основных принципов организации и разработки программ сложной структуры, этапов процесса объектно-ориентированного анализа и проектирования с применением языка UML, формирование у студентов навыков использования CASE-технологий в процессе анализа и разработки программных систем.

2. Место дисциплины в структуре образовательной программы

Дисциплина «Программная инженерия» относится к дисциплинам обязательной части образовательной программы.

Для освоения данной дисциплины студентам необходимы знания, полученные при изучении дисциплин «Алгоритмизация и программирование», «Информационные системы и технологии».

Знания и навыки, полученные при изучении дисциплины «Программная инженерия», используются учащимися при изучении последующих дисциплин, таких как «Проектирование информационных систем», «Разработка программных приложений», «Разработка корпоративных информационных систем», «Моделирование и проектирование государственных информационных систем»

3. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы.

Процесс изучения дисциплины направлен на формирование следующих элементов компетенций в соответствии с ФГОС ВО, ООП ВО и приобретения следующих знаний, умений, навыков и (или) опыта деятельности:

Формируемая компетенция (код и формулировка)	Индикатор достижения компетенции (код и формулировка)	Перечень планируемых результатов обучения
Общепрофессиональные компетенции		
ОПК-4 Способен участвовать в разработке стандартов, норм и правил, а также технической документации, связанной с профессиональной деятельностью	ИД-ОПК-4.1 Демонстрирует способность участвовать в разработке стандартов, норм и правил, а также технической документации, связанной с профессиональной деятельностью	Знать: <ul style="list-style-type: none">– основные стандарты, нормы и правила в области ИТ;– устройство, назначение и возможности применения CASE-технологий;– основные понятия методологии объектно-ориентированного анализа и проектирования. Уметь: <ul style="list-style-type: none">– собирать детальную информацию для формализации требований пользователей (заказчика);– применять CASE-технологии в процессе разработки технической документации. Владеть: <ul style="list-style-type: none">– навыками работы в CASE-системе;– навыками работы с научно-технической литературой и технической документацией по программному обеспечению.

<p>ОПК-8 Способен принимать участие в управлении проектами создания информационных систем на стадиях жизненного цикла</p>	<p>ИД-ОПК-8.1 Демонстрирует способность принимать участие в управлении проектами создания информационных систем на стадиях жизненного цикла;</p>	<p>Знать:</p> <ul style="list-style-type: none"> – основные понятия ООАП; – устройство, назначение и возможности применения CASE-технологий; – основные понятия и возможности языка UML, используемые в рамках объектно-ориентированного подхода к разработке программных систем; – устройство, назначение и возможности применения CASE-технологий. <p>Уметь:</p> <ul style="list-style-type: none"> – проектировать ИС в соответствии с профилем подготовки по видам обеспечения; – разрабатывать объектно-ориентированные модели, с использованием языка UML; – применять CASE-технологии в процессе анализа и разработки программных систем; <p>Владеть:</p> <ul style="list-style-type: none"> – навыками разработки объектно-ориентированных моделей программных систем и бизнес процессов; – навыками работы в CASE-системе.
--	---	---

4. Объем структура и содержание дисциплины «Программная инженерия»

Общая трудоемкость дисциплины составляет 4 зачетные единицы, 144 часа.

№ п/п	Темы (разделы) дисциплины, их содержание	Семестр	Виды учебных занятий, включая самостоятельную работу студентов, и их трудоемкость (в академических часах)						Формы текущего контроля успеваемости Форма промежуточной аттестации (по семестрам)
			Контактная работа						
			лекции	практические	лабораторные	консультации	аттестационные испытания	самостоятельная работа	
1	Основные понятия ООАП	5	2		2			2	Задания для самостоятельной работы
2	Понятие CASE- технологии	5	2		2			2	Задания для самостоятельной работы
3	Определение и структура языка UML	5	2		2			2	Задания для самостоятельной работы
4	Определение функциональных требований к системе.	5	4		4			6	Задания для самостоятельной работы

5	Моделирование классов.	5	6		6	3		6	Задания для самостоятельной работы Контрольная работа № 1
	<i>в том числе с ЭО и ДОТ</i>							2	Тест для самопроверки ЭУК в LMS Moodle
6	Моделирование взаимодействий. Диаграммы последовательности.	5	6		6			4	Задания для самостоятельной работы
7	Диаграммы состояний	5	4		4			4	Задания для самостоятельной работы
8	Диаграммы деятельности	5	4		4	3		2	Задания для самостоятельной работы Контрольная работа №2
	<i>в том числе с ЭО и ДОТ</i>							2	Тест для самопроверки ЭУК в LMS Moodle
9	Физическое представление модели	5	4		4			2	Задания для самостоятельной работы
						2	0,5	33,5	экзамен
	Всего за 5 семестр 144 часа		34		34	8	0,5	67,5	
	<i>в том числе с ЭО и ДОТ</i>							4	
	ИТОГО		34		34	8	0,5	67,5	
	<i>в том числе с ЭО и ДОТ</i>							4	

Содержание разделов дисциплины:

1. Основные понятия ООАП.

- 1.1. Основные принципы объектно-ориентированного анализа и проектирования.
- 1.2. Понятия модели, объекта, класса

2. Понятие CASE-технологии.

- 2.1. Общая характеристика и классификация CASE-средств.
- 2.2. Структура и функции CASE-систем.
- 2.3. Основные методологии разработки программных продуктов

3. Определение и структура языка UML.

- 3.1. История развития языка UML.
- 3.2. Определение, семантика и нотация языка UML.
- 3.3. Понятие диаграммы. Канонические диаграммы языка UML.

4. Определение функциональных требований к системе.

- 4.1. Диаграмма вариантов использования.
- 4.2. Дополнительные спецификации вариантов использования.
- 4.3. Применение CASE-системы для проектирования и документирования вариантов использования.

5. Моделирование классов.

- 5.1. Принципы анализа и разработки классов.
- 5.2. Диаграмма классов.

- 5.3. Отношения между классами.
- 5.4. Особенности реализации модели классов. Пакеты.
- 5.6. Применение CASE-системы для проектирования и документирования диаграмм классов.

6. Моделирование взаимодействий.

- 6.1. Диаграмма последовательности. Основные понятия.
- 6.2. Моделирование альтернативных потоков управления.
- 6.3. Применение CASE-системы для проектирования и документирования диаграмм последовательности.

7. Диаграмма состояний.

- 7.1. Основные понятия и особенности разработки диаграмм состояния.
- 7.2. Применение CASE-системы для проектирования и документирования диаграмм состояния.

8. Диаграмма деятельности.

- 8.1. Основные понятия и особенности разработки диаграмм состояния.
- 8.2. Применение CASE-системы для проектирования и документирования диаграмм деятельности.

9. Физическое представление модели.

- 9.1. Диаграммы компонентов и развертывания. Основные понятия и особенности разработки.
- 9.2. Применение CASE-системы для проектирования и документирования диаграмм деятельности.

5. Образовательные технологии, в том числе технологии электронного обучения и дистанционные образовательные технологии, используемые при осуществлении образовательного процесса по дисциплине

В процессе обучения используются следующие образовательные технологии:

Вводная лекция – дает первое целостное представление о дисциплине и ориентирует студента в системе изучения данной дисциплины. Студенты знакомятся с назначением и задачами курса, его ролью и местом в системе учебных дисциплин и в системе подготовки в целом. Дается краткий обзор курса, история развития науки и практики, достижения в этой сфере, имена известных ученых, излагаются перспективные направления исследований. На этой лекции высказываются методические и организационные особенности работы в рамках данной дисциплины, а также дается анализ рекомендуемой учебно-методической литературы.

Академическая лекция с элементами лекции-беседы – последовательное изложение материала, осуществляемое преимущественно в виде монолога преподавателя. Элементы лекции-беседы обеспечивают контакт преподавателя с аудиторией, что позволяет привлекать внимание студентов к наиболее важным темам дисциплины, активно вовлекать их в учебный процесс, контролировать темп изложения учебного материала в зависимости от уровня его восприятия.

Лабораторная работа – организация учебной работы с реальными материальными и информационными объектами, экспериментальная работа с аналоговыми моделями реальных объектов.

Консультации – вид учебных занятий, являющийся одной из форм контроля самостоятельной работы студентов. На консультациях по просьбе студентов рассматриваются наиболее сложные моменты при освоении материала дисциплины, преподаватель отвечает на вопросы студентов, которые возникают у них в процессе самостоятельной работы.

В процессе обучения используются следующие технологии электронного обучения и дистанционные образовательные технологии:

Электронный учебный курс «Программная инженерия» в LMS Электронный университет Moodle ЯрГУ, в котором:

- представлены задания для самостоятельной работы обучающихся по темам дисциплины;
- осуществляется проведение отдельных мероприятий текущего контроля успеваемости студентов;
- представлены видео лекций и теоретические материалы по отдельным темам дисциплины;
- представлены правила прохождения промежуточной аттестации по дисциплине;
- представлена основная учебная литература, рекомендуемая для освоения дисциплины.

6. Перечень лицензионного и (или) свободно распространяемого программного обеспечения, используемого при осуществлении образовательного процесса по дисциплине

В процессе осуществления образовательного процесса по дисциплине используются:

- электронные презентации для проведения лекций;
- программы Microsoft Office для формирования текстов материалов для промежуточной и текущей аттестации;
- для проведения лабораторных занятий используется CASE-среда Enterprise Architect (разработчик Sparx Systems).

7. Перечень современных профессиональных баз данных и информационных справочных систем, используемых при осуществлении образовательного процесса по дисциплине (при необходимости)

В процессе осуществления образовательного процесса по дисциплине используются:

- автоматизированная библиотечно-информационная система «БУКИ-NEXT»
http://www.lib.uniyl.ac.ru/opac/bk_cat_find.php

8. Перечень основной и дополнительной учебной литературы, ресурсов информационно-телекоммуникационной сети «Интернет» (при необходимости), рекомендуемых для освоения дисциплины

а) основная литература:

1. Ю.А. Ларина. Основы объектно-ориентированного моделирования с использованием языка UML. Ярославль, ЯрГУ, 2010.

б) дополнительная литература:

1. Леоненков А. В. Объектно-ориентированный анализ и проектирование с использованием UML и IBM RATIONAL ROSE: учеб. пособие для вузов. Бином. Лаборатория знаний, 2006.
2. Буч, Г. Язык UML. Руководство пользователя / Г. Буч, Дж.Рамбо, А. Якобсон – М.; ДМК, 2000.
3. Парамонов, И. В., Инженерия программных систем и комплексов на основе гибкой методологии разработки [Электронный ресурс] : учеб.-метод. пособие для студентов, обучающихся по направлению Прикладная математика и информатика / И. В. Парамонов; Яросл. гос. ун-т, Ярославль, ЯрГУ, 2015, 47с
4. Заботина Н. Н. Проектирование информационных систем: учеб. пособие для вузов. - М.: ИНФРА-М, 2013.
5. Лафоре Р. Объектно-ориентированное программирование в C++. / Р. Лафоре; [пер. с англ. А. Кузнецова, М. Назарова, В. Шрага] - 4-е изд. - СПб.: Питер, 2014.

9. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине

Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине включает в свой состав специальные помещения:

- учебные аудитории, для проведения занятий лекционного типа, оборудованные средствами проведения лекций в форме электронных презентаций;
- компьютерные классы с установленной ОС Windows XP и CASE-средой Enterprise Architect;
- учебные аудитории для проведения текущего контроля и промежуточной аттестации;
- помещения для самостоятельной работы;
- помещения для хранения и профилактического обслуживания технических средств обучения.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа к электронной информационно-образовательной среде ЯрГУ.

Автор(ы) :

Старший преподаватель кафедры ВПС Ю.А. Ларина

**Приложение №1 к рабочей программе дисциплины
«Программная инженерия»**

**Фонд оценочных средств
для проведения текущего контроля успеваемости
и промежуточной аттестации студентов
по дисциплине**

**1. Типовые контрольные задания или иные материалы,
используемые в процессе текущего контроля успеваемости**

Задания для самостоятельной работы
*(данные задания выполняются студентом самостоятельно
и преподавателем в обязательном порядке не проверяются)*

Задания по теме № 1 «Основные понятия ООАП» :

Глава № 1, раздел 1.1 учебного пособия «Основы объектно-ориентированного моделирования с использованием языка UML» (Ю.А. Ларина/ ЯрГУ, 2010).

Задания по теме № 2 «Понятие CASE-технологии»:

Выполнить анализ рынка современных CASE – систем.

Задания по теме № 3 «Определение и структура языка UML»:

Глава № 1, раздел 1.2-1.3 учебного пособия «Основы объектно-ориентированного моделирования с использованием языка UML» (Ю.А. Ларина/ ЯрГУ, 2010).

Задания по теме № 4 «Определение функциональных требований к системе»:

Упражнения № 1-3 после главы № 4 учебного пособия «Основы объектно-ориентированного моделирования с использованием языка UML» (Ю.А. Ларина/ ЯрГУ, 2010).

Задания по теме № 5 «Моделирование классов»:

Упражнения № 1-10 после главы № 2 учебного пособия «Основы объектно-ориентированного моделирования с использованием языка UML» (Ю.А. Ларина/ ЯрГУ, 2010).

Задания по теме № 6 «Моделирование взаимодействий»:

Упражнения № 4-6 после главы № 4 учебного пособия «Основы объектно-ориентированного моделирования с использованием языка UML» (Ю.А. Ларина/ ЯрГУ, 2010).

Задания по теме № 7 «Диаграмма состояний»:

Упражнения № 1-9 после главы № 3 учебного пособия «Основы объектно-ориентированного моделирования с использованием языка UML» (Ю.А. Ларина/ ЯрГУ, 2010).

Задания по теме № 8 «Диаграмма деятельности»:

Упражнения № 7-10 после главы № 4 учебного пособия «Основы объектно-ориентированного моделирования с использованием языка UML» (Ю.А. Ларина/ ЯрГУ, 2010).

Задания по теме № 9 «Физическое представление модели» :

Глава № 9 учебного пособия «Основы объектно-ориентированного моделирования с использованием языка UML» (Ю.А. Ларина/ ЯрГУ, 2010).

Тест для самопроверки **(тест проводится в ЭУК «Программная инженерия» в LMS Moodle)**

В тесте представлены задания на проверку знаний по пройденным разделам курса. Тест содержит теоретические и прикладные вопросы. В тесте 30 вопросов.

Максимальный балл за правильный ответ составляет 1 балл. Максимальное количество баллов, которое можно получить по итогам теста, составляет 30 баллов. Вопросы выбираются из банка вопросов случайным образом. На каждый вопрос дается одна попытка ответа. Студентам предоставляется две попытки прохождения теста. На прохождение теста дается 1 час.

Итоги прохождения теста оцениваются по следующим правилам:

- количество набранных баллов от 28 до 30 соответствует оценке «отлично»;
- количество набранных баллов от 25 до 27 соответствует оценке «хорошо»;
- количество набранных баллов от 20 до 24 соответствует оценке «удовлетворительно»;
- количество баллов меньше 20 соответствует оценке «неудовлетворительно».

Примерные вопросы теста

Вопрос 1. Объектно-ориентированный подход – это

- 1) технология создания сложного программного обеспечения, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определенного типа (класса);
- 2) технология создания сложного программного обеспечения, позволяющая вести практически независимую разработку отдельных частей (объектов) программы;
- 3) технология создания сложного программного обеспечения, основанная на представлении задачи исследования как объекта;
- 4) технология создания сложного программного обеспечения, основанная на объектном представлении кода программы;
- 5) технология создания сложного программного обеспечения, предназначенного для автоматизации технологических объектов.

Вопрос 2. CASE-технология – это

- 1) средство визуального моделирования, предназначенное для описания, визуализации и документирования объектно-ориентированных систем и бизнес-процессов с ориентацией на их последующую реализацию в виде программного обеспечения;
- 2) программное средство, поддерживающее процессы жизненного цикла ПО, включая анализ требований к системе, проектирование прикладного ПО и баз данных, генерацию кода, тестирование, документирование, обеспечение качества, управление конфигурацией ПО и управление проектом;
- 3) совокупность методологии проектирования информационных систем и набора инструментальных средств, позволяющих в наглядной форме анализировать и моделировать предметную область;
- 4) объектно-ориентированная методология проектирования информационных систем.

Вопрос 3. Моделирование в UML позволяет решать задачи

- 1) анализа и синтеза систем управления;
- 2) разработать и отладить программное обеспечение;
- 3) визуализировать систему в ее текущем или желательном для нас состоянии;
- 4) провести тестирование разработанного программного обеспечения;
- 5) описать структуру или поведение системы;
- 6) получить шаблон, позволяющий сконструировать систему;
- 7) документировать принимаемые решения, используя полученные модели.

Вопрос 4. Выделение характеристик и свойств объекта, которые позволяют его однозначно отделить от других объектов, называется

- 1) абстрагирование;
- 2) наследование;
- 3) инкапсуляция;
- 4) полиморфизм.

Вопрос 5. Возможность объектов с одинаковой спецификацией иметь различную реализацию называется

- 1) абстрагирование;
- 2) наследование;
- 3) инкапсуляция;
- 4) полиморфизм.

Правильные ответы

Вопрос №	Вариант ответа
1	1,2
2	3
3	3,6,7
4	1
5	4

Контрольная работа № 1

1. Разработать диаграмму вариантов использования для электронных часов с будильником. Часы отображают время, позволяют использовать будильник, переводить на нужные значения часы и минуты для установки текущего времени и времени будильника. Дополнительно часы имеют встроенный радиоприемник. Дать определения всем элементам графической нотации языка UML, которые были использованы для выполнения задания.
2. Разработать диаграмму классов для следующей задачи. Человек может быть подписан на несколько журналов, а на один журнал может быть подписано несколько человек. Каждый подписчик имеет имя и адрес, а журнал – название и стоимость подписки. Для подписки необходимо отслеживать дату и размер каждого платежа, а так же текущий срок окончания подписки. На разработанной диаграмме каждый класс должен содержать атрибуты, а возможно и операции, которые необходимо выделить из контекста задачи. Так же должны быть указаны кратности полюсов для всех отношений, представленных на диаграмме. Дать определения всем элементам графической нотации языка UML, которые были использованы для выполнения задания.

Контрольная работа № 2

1. Разработайте диаграмму состояний для следующей задачи. Простейшие цифровые часы могут работать в двух режимах: отображения и настройки. В режиме отображения часы показывают текущее время. Режим настройки состоит из настройки часов и настройки минут. Кнопка А позволяет выбрать нужный режим. Каждый раз при ее нажатии происходит переход к очередному режиму в последовательности: отображение, установка часов, установка минут, отображение и.т.п. Кнопка В позволяет увеличивать

значение часов или минут на единицу при каждом нажатии кнопки в соответствующем режиме установки, при этом отображается измененное значение часов или минут.

2. Разработайте диаграмму последовательности для входа в систему электронной почты, при котором ввод пароля и имени пользователя могут осуществляться в произвольном порядке. При неверно введенных имени или пароле, ввод повторяется до тех пор пока все не будет введено верно или пользователь не захочет выйти из системы.
3. Разработайте диаграмму деятельности для следующей задачи. Имеется 8 с виду одинаковых монет. Одна из них фальшивая и известно, что она легче настоящей. Как с помощью всего лишь двух взвешиваний на лабораторных весах, которые показывают только больше-меньше, найти фальшивую монету?

Правила выставления оценки по результатам контрольной работы:

Оценка по результатам контрольной работы определяется по следующему принципу:

- оценка «отлично» выставляется за правильно и полностью выполненное решение всех заданий контрольной работы;
- оценка «хорошо» выставляется, если решены все задания контрольной работы, но в решении одной из задач допущены несущественные ошибки;
- оценка «удовлетворительно» выставляется, если в решении более чем одной задачи допущены несущественные ошибки;
- оценка «неудовлетворительно» выставляется, если не решена одна из задач или в решении каждой из задач допущены существенные ошибки.

2. Список вопросов и (или) заданий для проведения промежуточной аттестации

Список вопросов к экзамену

1. Понятие CASE-технологии.
2. Общая характеристика и классификация CASE-средств.
3. Структура и функции CASE-систем.
4. Основные методологии разработки программных продуктов.
5. Основные принципы объектно-ориентированного анализа и проектирования.
6. Понятия модели, объекта, класса
7. Определение и структура языка UML.
8. История развития языка UML.
9. Понятие диаграммы. Канонические диаграммы языка UML.
10. Определение функциональных требований к системе. Диаграмма вариантов использования.
11. Дополнительные спецификации вариантов использования.
12. Моделирование классов. Принципы анализа и разработки классов. Диаграмма классов.
13. Отношения между классами. Особенности реализации модели классов.
14. Пакеты.
15. Моделирование взаимодействий. Диаграмма последовательности.
16. Моделирование альтернативных потоков управления.
17. Диаграмма состояний. Основные понятия и особенности разработки диаграмм состояний.
18. Диаграмма деятельности. Основные понятия и особенности разработки диаграмм деятельности.
19. Физическое представление модели. Диаграммы реализации.
20. Диаграммы компонентов. Основные понятия и особенности разработки диаграммы компонентов.

21. Диаграммы развертывания. Основные понятия и особенности разработки диаграммы развертывания.

Правила выставления оценки на экзамене.

В экзаменационный билет включается теоретический вопрос и две задачи на построение диаграмм. Примеры задач представлены в контрольных работах. На подготовку к ответу дается не менее 1 часа.

По итогам экзамена выставляется одна из оценок: «отлично», «хорошо», «удовлетворительно» или «неудовлетворительно».

Оценка «Отлично» выставляется студенту, который демонстрирует глубокое и полное владение содержанием материала и понятийным аппаратом объектно-ориентированного моделирования и языка UML; осуществляет межпредметные связи; умеет связывать теорию с практикой. Студент дает развернутые, полные и четкие ответы на вопрос экзаменационного билета и дополнительные вопросы, соблюдает логическую последовательность при изложении материала. Грамотно строит диаграммы по условию задач из билета и свободно использует терминологию ООАП.

Оценка «Хорошо» выставляется студенту, ответ которого на экзамене в целом соответствуют указанным выше критериям, но отличается меньшей обстоятельностью, глубиной, обоснованностью и полнотой. В решении задач имеют место отдельные неточности (несущественные ошибки), которые исправляются самим студентом после дополнительных и (или) уточняющих вопросов экзаменатора.

Оценка «Удовлетворительно» выставляется студенту, который дает недостаточно полные и последовательные ответы на вопросы экзаменационного билета и дополнительные вопросы, но при этом демонстрирует умение выделить существенные и несущественные признаки и установить причинно-следственные связи. Ответы излагаются в терминах ООАП и языка UML, но при этом допускаются ошибки в определении и раскрытии некоторых основных понятий. В решении задач допускает отдельные ошибки, большинство из которых исправляются самим студентом после дополнительных и (или) уточняющих вопросов экзаменатора.

Оценка «Неудовлетворительно» выставляется студенту, который демонстрирует разрозненные, бессистемные знания; беспорядочно и неуверенно излагает материал; не умеет выделять главное и второстепенное, не умеет соединять теоретические положения с практикой, не устанавливает межпредметные связи; допускает грубые ошибки при определении сущности раскрываемых понятий, вследствие непонимания их существенных и несущественных признаков и связей; дает неполные ответы, логика и последовательность изложения которых имеют существенные и принципиальные нарушения, в ответах отсутствуют выводы. Дополнительные и уточняющие вопросы экзаменатора не приводят к коррекции ответов студента. На основную часть дополнительных вопросов студент затрудняется дать ответ или дает неверные ответы. В решении задач допускает ошибки, которые студент затрудняется исправить самостоятельно после дополнительных и (или) уточняющих вопросов экзаменатора.

Оценка «Неудовлетворительно» выставляется также студенту, который взял экзаменационный билет, но отвечать отказался.

Приложение №2 к рабочей программе дисциплины «Программная инженерия»

Методические указания для студентов по освоению дисциплины

Основной формой изложения учебного материала по дисциплине «Программная инженерия» являются лекции, проводимые в виде электронных презентаций, что позволяет сделать материал лекций более наглядными, улучшает информативность и понимание изучаемого курса.

По большинству тем предусмотрены лабораторные работы, на которых происходит закрепление лекционного материала путем применения его к конкретным задачам и получение навыков работы непосредственно с CASE- системой, путем выполнения заданий по построению моделей определенного типа. Выполнение предлагаемых в процессе изучения курса лабораторных работ позволяет не только понять и закрепить теоретический материал, но и приобрести навык анализа предметной области и построения объектно-ориентированных моделей с применением современных технологий.

Для успешного освоения дисциплины очень важно решение достаточно большого количества задач, как в аудитории, так и самостоятельно в качестве домашних заданий. Примеры решения задач разбираются и обсуждаются на лекциях и лабораторных занятиях. Основная цель решения задач – помочь усвоить фундаментальные понятия и основы концепции объектно-ориентированного анализа и проектирования. Для решения всех задач необходимо знать и понимать лекционный материал. Поэтому в процессе изучения дисциплины рекомендуется регулярное повторение пройденного лекционного материала. Материал, законспектированный на лекциях, необходимо дома еще раз прорабатывать и при необходимости дополнять информацией, полученной при выполнении лабораторных работ или из учебной литературы.

Большое внимание должно быть уделено выполнению домашней работы. В качестве заданий для самостоятельной работы дома студентам предлагаются задачи, аналогичные разобранным на лекциях и лабораторных занятиях или немного более сложные, которые являются результатом объединения нескольких базовых задач.

Для проверки и контроля усвоения теоретического материала, приобретенных практических навыков работы с CASE - системой, в течение обучения проводятся мероприятия текущей аттестации в виде двух контрольных работ. Для более успешной подготовки к контрольным работам перед их проведением преподавателем проводятся консультации.

В конце семестра студенты сдают экзамен. Экзамен принимается по экзаменационным билетам, каждый из которых включает в себя теоретический вопрос и две задачи. На самостоятельную подготовку к экзамену выделяется три дня, в это время предусмотрена групповая консультация.

Освоить вопросы, излагаемые в процессе изучения дисциплины «Программная инженерия» самостоятельно студенту достаточно сложно. Это связано с отсутствием опыта в проведении анализа предметной области и построения объектно-ориентированных моделей. Поэтому посещение всех аудиторных занятий является совершенно необходимым. Без упорных и регулярных занятий в течение семестра сдать зачет по итогам изучения дисциплины студенту будет сложно.