

перел

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ**

УТВЕРЖДАЮ
Декан ФВТ  Л.Р. Фионова
« 23 » сентября 2016 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

С1.2.6 Системное программное обеспечение

Специальность: 09.05.01 *«Применение и эксплуатация автоматизированных систем специального назначения»*

Специализация №12: *«Автоматизированные системы обработки информации и управления специального назначения»*

Квалификация (степень) выпускника: *инженер*

Форма обучения: *очная*

Пенза, 2016

Цели освоения дисциплины

Целями освоения дисциплины «Системное программное обеспечение» являются: подготовка студента к самостоятельному выполнению работ по созданию автоматизированных систем обработки информации и управления специального назначения, соответствующих общекультурным, общепрофессиональным, профессиональным и профессионально-специализированным компетенциям; овладение студентами знаниями и навыками в области разработки приложений.

2. Место дисциплины в структуре ОПОП специалитета

Учебная дисциплина «Системное программное обеспечение» относится к вариативной части блока С1, шифр дисциплины С1.2.4.

Дисциплина опирается на знания, полученные студентами в ходе изучения дисциплин «Математика», «Информатика», «Программирование», «Технологии программирования», «Операционные системы».

Компетенции, приобретенные в ходе изучения дисциплины «Системное программное обеспечение», готовят студента к освоению профессиональных компетенций.

Знания, умения и навыки, полученные в результате изучения дисциплины, найдут применение при изучении следующих дисциплин: «Моделирование и проектирование систем», «Программирование в компьютерных сетях», «Системы реального времени», «Проектирование автоматизированных систем специального назначения», а также при выполнении курсовых и дипломного проектов.

3. Компетенции обучающегося, формируемые в результате освоения дисциплины «Системное программное обеспечение»

Процесс изучения дисциплины направлен на формирование элементов следующих компетенций в соответствии с ФГОС ВО по данной специальности:

Коды компетенции	Наименование компетенции	Структурные элементы компетенции (в результате освоения дисциплины обучающийся должен знать, уметь, владеть)
ОПК-4	способность использовать языки и системы программирования, программные средства общего назначения, инструментальные средства компьютерного моделирования для решения различных исследовательских и профессиональных задач	Знать: методологии проектирования программных систем, канонические модели языка UML.
		Уметь: использовать основные концепции объектно-ориентированного проектирования; отлаживать и тестировать программы
		Владеть: основами разработки и отладки программных систем; разработкой типовых элементов программной документации; основами коллективной разработки программных изделий.

4. Структура и содержание дисциплины «Системное программное обеспечение»

4.1. Структура дисциплины

Общая трудоемкость дисциплины составляет 4 зачетных единиц, 144 часа.

№ п/п	Наименование разделов и тем дисциплины (модуля)	Семестр	Недели семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)						Формы текущего контроля успеваемости (по неделям семестра)	
				Аудиторная работа			Самостоятельная работа				
				Всего	Лекции	Лабораторные занятия	Всего	Подготовка к лабораторным занятиям	Курсовая работа	Защита лабораторных работ	Курсовая работа
1	Введение	6	1	1	1						
2	Раздел 1. Процесс создания программного обеспечения	6	1-2	2	2						
3	Раздел 2. Основы методологии объектно-ориентированного подхода	6	2-3	2	2					×	
4	Тема 2.1. Элементы объектного подхода	6	2		1						
5	Тема 2.2. Объекты и классы	6	3		1						
6	Раздел 3. Разработка требований к программным системам	6	3-4	15	3	12	8	8		×	
7	Тема 3.1. Составление технического задания на разработку	6	3		1						
8	Тема 3.2. Язык UML. Диаграммы вариантов использования	6	4		2						
9	Раздел 4. Объектно-ориентированное проектирование	6	5-9	44	10	34	20	20			

	программных систем									
10	Тема 4.1. Диаграммы классов	6	5		2					
11	Тема 4.2. Диаграммы последовательности	6	6		2					
12	Тема 4.3. Диаграммы деятельности	6	7		1				×	
13	Тема 4.4. Диаграммы состояний	6	7		1					
14	Тема 4.5. Диаграммы компонентов	6	8		1					
15	Тема 4.6. Диаграммы размещения	6	8		1					
	Тема 4.7. Паттерны проектирования	6	9		2					
16	Раздел 5. Управление ресурсами в жизненном цикле программных систем	6	10-11	3	3				×	
17	Тема 5.1. Структура проектной группы	6	10		1					
18	Тема 5.2. Основные виды ресурсов и управление ими	6	10-11		2					
19	Раздел 6. Верификация и тестирование программных систем	6	11-13	4	4					
20	Тема 6.1. Задачи и суть верификации программных систем	6	11		1					
21	Тема 6.2. Методы тестирования	6	12-13		3				×	
22	Раздел 7. Документирование, внедрение и сопровождение программных систем	6	13-14	8	2	6	8	8		
23	Тема 7.1. Разработка пользовательской документации	6	13		1					
24	Тема 7.2. Поддержка программных систем	6	14		1					
25	Раздел 8. Управление процессом разработки	6	14-16	4	4					
26	Тема 8.1. Планирование проекта	6	14-15		2				×	
27	Тема 8.2. Командная работа, распределение ролей и ответственности	6	15-16		2					
28	Раздел 9. Методологии создания программных решений	6	16-17	2	2					
29	Тема 9.1. Методология MSF	6	16		1					
30	Тема 9.2. Agile – методологии	6	17		1				×	

4.2. Содержание дисциплины

4.2.1. Содержание лекций

Введение

Классификация программного обеспечения (ПО). Требования и показатели качества ПО. Предмет программной инженерии. Структура и содержание дисциплины.

1. Процесс создания программного обеспечения

Понятие жизненного цикла программного изделия. Модели жизненного цикла. Процессы жизненного цикла согласно ГОСТ Р ИСО/МЭК 12207 – 2010.

2. Основы методологии объектно-ориентированного подхода

Элементы объектного подхода. Абстрагирование. Ограничение доступа. Модульность. Иерархия. Типизация. Параллелизм. Устойчивость.

Состояние объекта. Поведение объекта. Индивидуальность объекта. Интерфейс объекта. Время жизни объекта. Отношения между объектами. Отношение использования. Отношение включения. Посылка сообщений.

Понятие класса. Отношения между классами. Наследование. Простое наследование. Простой полиморфизм. Множественное наследование. Множественный полиморфизм. Включение. Ассоциация. Использование. Наполнение.

3. Разработка требований к программным системам

Составление технического задания на разработку. Унифицированный язык моделирования (UML). Диаграммы вариантов использования (Use Case диаграммы).

4. Объектно – ориентированное проектирование программных систем

Задачи проектирования. Идентификация классов и объектов. Основные результаты объектно-ориентированного проектирования (в стандарте UML): диаграммы классов, объектов, последовательности, деятельности, состояний, компонентов, размещения. CASE – инструменты. Концептуальный, логический и физический дизайн. Прототипирование. Паттерны проектирования.

5. Управление ресурсами в жизненном цикле программных систем

Структура проектной группы. Ресурсы для обеспечения функциональной пригодности ПО. Ресурсы на реализацию характеристик качества ПО. Ресурсы на имитацию внешней среды для обеспечения испытаний программной системы.

6. Верификация и тестирование программных систем

Верификация. Методы тестирования программных систем. Планы тестирования проекта, компонентов. Модели анализа тестов. Автоматизированные регрессивные тесты. Тестирование классов и иерархий классов. Тестирование распределенных объектов. Тестирование интернет-приложений.

7. Документирование, внедрение и сопровождение программных систем

Разработка пользовательской документации. Создание программы установки. Обучение пользователей, поддержка ПО. Повторное использование и переносимость ПО, реинженерия.

8. Управление процессом разработки

Планирование проекта. Анализ рисков. Оценка затрат. Внутренняя проектная документация. Командная работа, распределение ролей и ответственности (централизованный, децентрализованный и смешанный тип управления). Диаграммы Ганта. Графики PERT. Контроль качества. Анализ текущего состояния проекта. Стабильность/нестабильность проекта, построение графиков интенсивности возникновения и устранения ошибок. Средства поддержки управления проектом. Модель развития функциональных возможностей CMM.

9. Методологии создания программных решений

Основные концепции методологии MICROSOFT SOLUTIONS FRAMEWORK (MSF): модель процессов, управление проектом, модель проектной группы, управление рисками. Позиционирование MSF в сравнении с другими методологиями разработки программного обеспечения. Инструментальная поддержка методологии MSF в среде разра-

ботки Microsoft Visual Studio.

Быстрая разработка приложений (RAD).

Rational Unified Process.

Гибкие (agile) методы разработки программных систем. Метод Scrum.

Заключение

Основные тенденции развития методов и средств программной инженерии.

4.2.2. Перечень и содержание лабораторных занятий

Основные цели лабораторного практикума – привитие студентам навыков в проектировании программных систем на основе методологии объектного подхода. Тематика задач, решаемых на лабораторных занятиях, определяется содержанием соответствующих разделов рабочей программы. Содержание задач увязывается со спецификой будущей специальности. Форма проведения лабораторного практикума – решение задач, связанных с выполнением отдельных этапов проектирования программных систем.

№ п/п	№ раздела	Наименование лабораторных работ	Кол. часов
1	3	Изучение работы в среде одного из CASE-средств, поддерживающих язык UML (IBM Rational Rose, ArgoUML, StarUML, Visual-Paradigm for UML)	4
2	3	Разработка диаграммы вариантов использования системы	8
3	4	Разработка диаграмм классов системы	8
4	4	Разработка диаграмм последовательности	8
5	4	Разработка диаграмм состояний	6
6	4	Разработка диаграммы компонентов системы	6
7	4	Разработка диаграммы размещения системы	6
8	7	Разработка комплекта эксплуатационных документов для сложного программного продукта	6

4.2.3. Содержание курсового проектирования

Курсовое проектирование заключается в разработке и отладке многомодульного приложения. Разработка ведется с использованием CASE-средств в стандарте языка UML, реализация – в среде Microsoft Visual C++, Microsoft Visual C# или Embarcadero (CodeGear) C++ Builder.

Темы курсовых работ

1. Учебная СУБД
2. Учебная операционная оболочка
3. Web – браузер
4. Справочная система “Маршруты городского транспорта”
5. Программа тестирования студентов

6. Редактор графических трехмерных объектов
7. Расписание занятий студентов и преподавателей
8. Редактор схем алгоритмов
9. Мастер создания визитных карточек
10. Переводчик текстов
11. Клавиатурный тренажер
12. Учебный графический редактор
13. Органайзер
14. Системный будильник
15. Имитатор инсталлятора приложений
16. Имитатор почтовой системы
17. “Фоторобот”
18. Инженерный калькулятор
19. Сотовый телефон
20. Имитатор работы ЛВС
21. Имитатор дефрагментатора жестких дисков

5. Образовательные технологии

В процессе изучения дисциплины применяются следующие образовательные технологии:

- чтение лекций проводится с использованием мультимедийного компьютерного проектора с раздачей копий демонстрируемых слайдов;
- практически все лекции и часть лабораторных занятий проводятся в интерактивной форме с разбором конкретных моделей программных систем, что составляет примерно 30 – 40 % аудиторных занятий;
- мастер-классы по созданию различных моделей языка UML в среде одного из CASE-средств;
- при выполнении лабораторного практикума и во время самостоятельной работы используются обучающие программы с сайта кафедры ИВС (<http://ivs-pgy.nm.ru>) и Интернет-ресурсы.

6. Учебно-методическое обеспечение самостоятельной работы студентов. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины.

6.1. План самостоятельной работы студентов

№ нед.	Тема	Вид самостоятельной работы	Задание	Рекомендуемая литература	Кол-во часов
4	Разработка требований к программным системам	Подготовка к лабораторному занятию. Оформление отчета по лабораторной работе	Выделение действующих лиц и вариантов использования проектируемой программной системы. Разработка диаграммы вариантов использования программной системы.	/1, 2/	30

6	Объектно – ориентированное проектирование программных систем	Подготовка к лабораторному занятию. Оформление отчета по лабораторной работе	Создание основных классов и пакетов системы. Построение диаграмм классов.	/1, 2/	20
8	Объектно – ориентированное проектирование программных систем	Подготовка к лабораторному занятию. Оформление отчета по лабораторной работе	Изучение взаимодействия объектов. Построение диаграмм последовательности для каждого варианта использования.	/1, 2/	20
10	Объектно – ориентированное проектирование программных систем	Подготовка к лабораторному занятию. Оформление отчета по лабораторной работе	Изучение поведения объектов. Построение диаграмм состояний для отдельных объектов системы.	/1, 2/	10
12	Объектно – ориентированное проектирование программных систем	Подготовка к лабораторному занятию. Оформление отчета по лабораторной работе	Изучение представления компонентов. Построение диаграммы компонентов системы.	/1, 2/	10
15	Объектно – ориентированное проектирование программных систем	Подготовка к лабораторному занятию. Оформление отчета по лабораторной работе	Изучение представления размещения системы. Построение диаграммы размещения системы.	/1, 2/	10
18	Документирование, внедрение и сопровождение программных систем	Подготовка к лабораторному занятию. Оформление отчета по лабораторной работе	Изучение организации документирования программных средств. Разработка комплекта эксплуатационных документов для программной системы.	/1, 2/	26

6.2. Методические указания по организации самостоятельной работы студентов

Планируются следующие виды самостоятельной работы:

- подготовка к лабораторным работам занятиям;
- оформление отчётов по лабораторным работам;
- работа по выполнению курсовой работы;
- подготовка к зачету;
- работа с конспектом лекций и изучение литературы при подготовке к экзаменам.

В качестве учебно-методического обеспечения по организации самостоятельной

работы студентов при подготовке к аудиторным занятиям рекомендуется использовать основную и дополнительную литературу, а также электронные методические указания по данной дисциплине, размещенные в университетской локальной сети. Также во время самостоятельной работы используются материалы сайта «Интернет Университет Информационных Технологий» (<http://www.intuit.ru>).

6.3. Материалы для проведения текущего и промежуточного контроля знаний студентов

Контроль освоения компетенций

№ п/п	Вид контроля	Контролируемые темы (разделы)	Компетенции, компоненты которых контролируются
1	Текущий: собеседование при защите лабораторных работ	Разделы 3, 4, 7, 8	ОПК-4
2	Промежуточный: защита курсовой работы	Разделы 3, 4, 7, 8	ОПК-4
3	Промежуточный: зачет (вопрос и задание)	Разделы 1 – 9	ОПК-4

Примерный перечень вопросов и заданий к зачету

Вопросы к зачету

1. Жизненный цикл (ЖЦ) программного изделия. Содержание основных этапов ЖЦ.
2. Каскадная модель разработки программной системы.
3. Эволюционная модель разработки программной системы.
4. Спиральная модель разработки программной системы.
5. Быстрая разработка приложений (RAD).
6. Rational Unified Process.
7. Гибкие (agile) методы разработки программных систем. Метод Scrum.
8. Основные концепции методологии создания программных решений Microsoft Solution Framework (MSF).
9. Объекты, их свойства (состояние, поведение, индивидуальность).
10. Объектно-ориентированный анализ: назначение, основные результаты.
11. Объектно-ориентированное проектирование: назначение, основные результаты.
12. Язык моделирования UML: назначение, основные модели, области применения.
13. Диаграмма вариантов использования языка UML (Use Case Diagram).
14. Диаграмма классов языка UML (Class Diagram).
15. Диаграмма последовательности языка UML (Sequence Diagram).
16. Диаграмма деятельности языка UML (Activity Diagram).
17. Диаграмма состояний языка UML (Statechart Diagram).
18. Диаграммы реализации языка UML: диаграмма компонентов (Component Diagram) и диаграмма размещения (Deployment Diagram).
19. Проектирование модулей программной системы.
20. Проектирование интерфейса пользователя.
21. Стиль программирования.
22. Планирование проекта и управление требованиями.
23. Ресурсы в программных проектах. Управление ресурсами.
24. Управление проектами. Организация проектной команды. Роли участников в программных проектах.
25. Средства поддержки управления проектом. Модель развития функциональных возможностей СММ.

26. Управление рисками в программных проектах: анализ, идентификация, ранжирование.
27. Управление рисками в программных проектах: планирование, разрешение, наблюдение.
28. Тестирование программной системы. Функциональное и структурное тестирование.
29. Организация тестирования программной системы. Тестирование элементов.
30. Организация тестирования программной системы. Интеграционное тестирование.
31. Организация тестирования программной системы. Системное тестирование.
32. Документирование разрабатываемой программной системы.
33. Выпуск программного продукта.
34. Качество программного обеспечения. Характеристики качества. Оценка качества программного обеспечения.

Задания к зачету

1. Система представляет собой текстовый редактор типа Блокнот. Разработать диаграмму вариантов использования этой системы. Разработать сценарий одного из вариантов использования.
2. Система представляет собой электронную таблицу типа Excel. Разработать диаграмму вариантов использования этой системы. Разработать сценарий одного из вариантов использования.
3. Система представляет собой текстовый процессор типа Word. Разработать диаграмму вариантов использования этой системы. Разработать сценарий одного из вариантов использования.
4. Система представляет собой приложение для подготовки презентаций типа Power Point. Разработать диаграмму вариантов использования этой системы. Разработать сценарий одного из вариантов использования.
5. Система представляет собой Web – браузер (типа Internet Explorer, Mozilla Firefox, Opera, Chrome и т.п.). Разработать диаграмму вариантов использования этой системы. Разработать сценарий одного из вариантов использования.
6. Система представляет собой реляционную СУБД типа MS Access. Разработать диаграмму вариантов использования этой системы. Разработать сценарий одного из вариантов использования.
7. Система представляет собой операционную оболочку типа Проводник Windows. Разработать диаграмму вариантов использования этой системы. Разработать сценарий одного из вариантов использования.
8. Система представляет собой графический редактор типа Paint ОС Windows. Разработать диаграмму вариантов использования этой системы. Разработать сценарий одного из вариантов использования.
9. Система представляет собой сотовый телефон. Разработать диаграмму вариантов использования этой системы. Разработать сценарий одного из вариантов использования.
10. Система представляет собой банкомат. Разработать диаграмму вариантов использования этой системы. Разработать сценарий одного из вариантов использования.
11. Система представляет собой файловый менеджер типа Total Commander. Разработать диаграмму вариантов использования этой системы. Разработать сценарий одного из вариантов использования.
12. Приложение представляет собой почтовую систему (типа MS Outlook Express или Mail.ru). Разработать диаграмму вариантов использования этой системы. Разработать сценарий одного из вариантов использования.
13. Система представляет собой клавиатурный тренажер ПК. Разработать диаграмму вариантов использования этой системы. Разработать сценарий одного из вариантов ис-

пользования.

14. Система представляет собой телефонный справочник. Разработать диаграмму вариантов использования этой системы. Разработать сценарий одного из вариантов использования.

15. Система представляет собой переводчик текстов. Разработать диаграмму вариантов использования этой системы. Разработать сценарий одного из вариантов использования.

16. Система представляет собой телевизор. Разработать диаграмму вариантов использования этой системы. Разработать сценарий одного из вариантов использования.

7. Учебно-методическое и информационное обеспечение дисциплины

а) основная литература

1. Антамошкин, О.А. Программная инженерия. Теория и практика: учебник [Электронный ресурс]/ О.А. Антамошкин. – Красноярск: СФУ, 2012. – 247 с. – Режим доступа: <https://e.lanbook.com/book/45709>. – Электрон. версия печ. публикации. – ЭБС Лань: требуется авторизация пользователя.

2. Батоврин, В.К. Системная и программная инженерия. Словарь-справочник: учеб. пособие для вузов [Электронный ресурс]: учеб. пособие для вузов [Гриф УМО] / В.К. Батоврин. – М.: ДМК Пресс, 2010. – 280 с. – Режим доступа: <http://e.lanbook.com/view/book/1097/>. – Электрон. версия печ. публикации. – ЭБС Лань: требуется авторизация пользователя.

б) дополнительная литература

3. Буч, Г. Язык UML. Руководство пользователя. 2-е изд. [Электронный ресурс] / Г. Буч, Д. Рамбо, И. Якобсон – М.: ДМК Пресс. – 496 с. Режим доступа: <http://e.lanbook.com/view/book/1246/>. – Электрон. версия печ. публикации. – ЭБС Лань: требуется авторизация пользователя.

4. ГОСТ Р ИСО/МЭК 12207-2010 Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств. Российская Федерация. Государственный стандарт от 01 марта 2012 года. – Режим доступа: <http://docs.cntd.ru/document/gost-r-iso-mek-12207-2010>.

в) программное обеспечение и Интернет-ресурсы:

1) Программные продукты:

- IBM Rational Rose 2000 (или ArgoUML, StarUML, Visual-Paradigm for UML),
- Microsoft Solutions Framework версии 4.0 или выше,
- Microsoft Visual C++ версии 2010 или выше, Microsoft Visual C# версии 2010 или выше, Embarcadero (CodeGear) C++ Builder.
- ОС Windows XP/Windows Vista/Windows 7/8/10 или Linux.

2) Интернет-ресурсы:

- Материалы раздела «Программная инженерия» сайта «Интернет – Университет Информационных Технологий», <http://www.intuit.ru/>
- ЭБС Лань
- ЭБС biblio-online.ru

8. Материально-техническое обеспечение дисциплины

Лекционные занятия проводятся в аудитории, оснащенной компьютерным проектором, проекционным экраном, шторами, сетью электропитания 220 В.

Лабораторные занятия проводятся в классе, оснащенный персональными компьютерами с операционной системой Windows XP/Windows Vista/Windows 7/8/10 или Linux.

Рабочая программа дисциплины «Системное программное обеспечение» составлена в соответствии с требованиями ФГОС ВО по специальности 09.05.01 – «Применение и эксплуатация автоматизированных систем специального назначения».

Программу составил:

Убиенных Г.Ф., доцент кафедры «Информационно-вычислительные системы»



Настоящая программа не может быть воспроизведена ни в какой форме без предварительного письменного разрешения кафедры-разработчика программы.

Программа одобрена на заседании кафедры «Информационно-вычислительные системы»
Протокол № 1 от « 06 » 09 2016 г.

Зав. кафедрой ИВС



Косников Ю.Н.

Программа одобрена методической комиссией факультета ВТ


Протокол № 1 от « 22 » 09 2016 г.

Председатель методической комиссии
факультета ВТ



Глотова Т.В.

Сведения о переутверждении программы на очередной учебный год и регистрации изменений

Учебный год	Решение кафедры (№ протокола, дата, подпись зав. кафедрой)	Внесенные изменения	Номера листов (страниц)		
			замененных	новых	аннулированных
2017 2018	Реш. № 14 от 27.06.17 	Внесены изменения информации на ЭБС и сайт университета	11		