

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
"Казанский (Приволжский) федеральный университет"  
Институт вычислительной математики и информационных технологий



**УТВЕРЖДАЮ**

Проректор  
по образовательной деятельности КФУ  
Проф. Таюрский Д.А.

\_\_\_\_\_ 20\_\_ г.

**Программа дисциплины**

Конструирование программного обеспечения Б1.Б.25

Направление подготовки: 09.03.04 - Программная инженерия

Профиль подготовки: Технологии разработки информационных систем

Квалификация выпускника: бакалавр

Форма обучения: очное

Язык обучения: русский

**Автор(ы):**

Прокопьев Н.А.

**Рецензент(ы):**

Гусенков А.М.

**СОГЛАСОВАНО:**

Заведующий(ая) кафедрой: Вахитов Г. З.

Протокол заседания кафедры No \_\_\_\_ от " \_\_\_\_ " \_\_\_\_\_ 201\_\_ г

Учебно-методическая комиссия Института вычислительной математики и информационных технологий:

Протокол заседания УМК No \_\_\_\_ от " \_\_\_\_ " \_\_\_\_\_ 201\_\_ г

Регистрационный No

## Содержание

1. Цели освоения дисциплины
2. Место дисциплины в структуре основной образовательной программы
3. Компетенции обучающегося, формируемые в результате освоения дисциплины /модуля
4. Структура и содержание дисциплины/ модуля
5. Образовательные технологии, включая интерактивные формы обучения
6. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины и учебно-методическое обеспечение самостоятельной работы студентов
7. Литература
8. Интернет-ресурсы
9. Материально-техническое обеспечение дисциплины/модуля согласно утвержденному учебному плану

Программу дисциплины разработал(а)(и) ассистент, б.с. Прокопьев Н.А. кафедра технологий программирования отделение фундаментальной информатики и информационных технологий, NiAProkopenv@kpfu.ru

### 1. Цели освоения дисциплины

Цель данной дисциплины - изучить и освоить подходы и способы конструирования современного программного обеспечения, дать навыки составления и тестирования программ в соответствии с созданными проектами и использованием интегрированных систем разработки программного обеспечения.

### 2. Место дисциплины в структуре основной образовательной программы высшего профессионального образования

Данная учебная дисциплина включена в раздел "Б1.Б.25 Дисциплины (модули)" основной образовательной программы 09.03.04 Программная инженерия и относится к базовой (общепрофессиональной) части. Осваивается на 3 курсе, 6 семестр.

Для успешного освоения данной дисциплины нужно освоение в качестве предшествующих следующих дисциплин: 'Информатика', 'Объектно-ориентированный анализ и программирование'.

### 3. Компетенции обучающегося, формируемые в результате освоения дисциплины /модуля

В результате освоения дисциплины формируются следующие компетенции:

Шифр компетенции	Расшифровка приобретаемой компетенции
ОПК-2 (профессиональные компетенции)	владением архитектурой электронных вычислительных машин и систем
ПК-13 (профессиональные компетенции)	готовность к использованию методов и инструментальных средств исследования объектов профессиональной деятельности
ПК-3 (профессиональные компетенции)	владение навыками использования различных технологий разработки программного обеспечения
ПК-4 (профессиональные компетенции)	владение концепциями и атрибутами качества программного обеспечения (надежности, безопасности, удобства использования), в том числе роли людей, процессов, методов, инструментов и технологий обеспечения качества
ПК-6 (профессиональные компетенции)	владением классическими концепциями и моделями менеджмента в управлении проектами
ПК-9 (профессиональные компетенции)	владением методами контроля проекта и готовностью осуществлять контроль версий

В результате освоения дисциплины студент:

1. должен знать:

- основы объектно-ориентированного подхода к разработке программного обеспечения;
- основы технологии разработки программного обеспечения;

- основы проектирования и архитектуры программного обеспечения.

2. должен уметь:

- ставить и решать прикладные задачи с использованием современных информационно-коммуникационных технологий;

- программировать на одном из алгоритмических языков;

3. должен владеть:

- навыками работы со средствами автоматизации разработки ПО;

- основами алгоритмизации и программирования.

4. должен демонстрировать способность и готовность:

- применять полученные знания и навыки в своей профессиональной деятельности.

#### 4. Структура и содержание дисциплины/ модуля

Общая трудоемкость дисциплины составляет 2 зачетных(ые) единиц(ы) 72 часа(ов).

Форма промежуточного контроля дисциплины: зачет в 6 семестре.

Суммарно по дисциплине можно получить 100 баллов, из них текущая работа оценивается в 50 баллов, итоговая форма контроля - в 50 баллов. Минимальное количество для допуска к зачету 28 баллов.

86 баллов и более - "отлично" (отл.);

71-85 баллов - "хорошо" (хор.);

55-70 баллов - "удовлетворительно" (удов.);

54 балла и менее - "неудовлетворительно" (неуд.).

#### 4.1 Структура и содержание аудиторной работы по дисциплине/ модулю

##### Тематический план дисциплины/модуля

N	Раздел Дисциплины/ Модуля	Семестр	Неделя семестра	Виды и часы аудиторной работы, их трудоемкость (в часах)			Текущие формы контроля
				Лекции	Практи- ческие занятия	Лабора- торные работы	
1.	Тема 1. Основные понятия конструирования ПО.	6		0	3	3	Письменное домашнее задание
2.	Тема 2. Проектирование при конструировании.	6		0	3	3	Письменное домашнее задание
3.	Тема 3. Методология конструирования программного кода.	6		0	3	3	Компьютерная программа
4.	Тема 4. Защитное программирование.	6		0	3	3	Компьютерная программа
5.	Тема 5. Создание качественного кода.	6		0	3	3	Отчет

N	Раздел Дисциплины/ Модуля	Семестр	Неделя семестра	Виды и часы аудиторной работы, их трудоемкость (в часах)			Текущие формы контроля
				Лекции	Практи- ческие занятия	Лабора- торные работы	
6.	Тема 6. Тестирование в конструировании.	6		0	3	3	Компьютерная программа
.	Тема . Итоговая форма контроля	6		0	0	0	Зачет
	Итого			0	18	18	

## 4.2 Содержание дисциплины

### Тема 1. Основные понятия конструирования ПО.

#### **практическое занятие (3 часа(ов)):**

Понятие конструирование программного обеспечения. Важность конструирования ПО. Основные этапы конструирования. Минимизация сложности. Ожидание изменений. Конструирование с возможностью проверки. Стандарты в конструировании.

#### **лабораторная работа (3 часа(ов)):**

Создание UML-диаграмм для планируемой программной системы

### Тема 2. Проектирование при конструировании.

#### **практическое занятие (3 часа(ов)):**

Принципы проектирования. Методики проектирования.

#### **лабораторная работа (3 часа(ов)):**

Проектирование классов планируемой программной системы

### Тема 3. Методология конструирования программного кода.

#### **практическое занятие (3 часа(ов)):**

Модели конструирования. Планирование конструирования. Измерения в конструировании. Документация, разрабатываемая при конструировании. Принципы конструирования. Способы конструирования. Создание классов и методов. Переменные и операторы. Табличные методы.

#### **лабораторная работа (3 часа(ов)):**

Реализация взаимодействия классов. Формирование подсистем создаваемой программной системы.

### Тема 4. Защитное программирование.

#### **практическое занятие (3 часа(ов)):**

Защита программы от неправильных входных данных. Способы обработки ошибок. Исключения. Изоляция повреждений, вызванных ошибками.

#### **лабораторная работа (3 часа(ов)):**

Рефакторинг и улучшение обработки ошибок в создаваемой программной системе

### Тема 5. Создание качественного кода.

#### **практическое занятие (3 часа(ов)):**

Характеристики качества кода. Методики повышения качества кода. Контроль качества кода.

#### **лабораторная работа (3 часа(ов)):**

Оценка качества созданной системы

### Тема 6. Тестирование в конструировании.

#### **практическое занятие (3 часа(ов)):**

Модульное тестирование. Разработка, управляемая тестированием. Интеграционное тестирование.

### лабораторная работа (3 часа(ов)):

Написание модульных тестов для созданной системы

### 4.3 Структура и содержание самостоятельной работы дисциплины (модуля)

N	Раздел дисциплины	Семестр	Неделя семестра	Виды самостоятельной работы студентов	Трудоемкость (в часах)	Формы контроля самостоятельной работы
1.	Тема 1. Основные понятия конструирования ПО.	6		подготовка домашнего задания	6	Письменное домашнее задание
2.	Тема 2. Проектирование при конструировании.	6		подготовка домашнего задания	6	Письменное домашнее задание
3.	Тема 3. Методология конструирования программного кода.	6			6	Компьютерная программа
4.	Тема 4. Защитное программирование.	6			6	Компьютерная программа
5.	Тема 5. Создание качественного кода.	6		подготовка к отчету	6	Отчет
6.	Тема 6. Тестирование в конструировании.	6			6	Компьютерная программа
	Итого				36	

### 5. Образовательные технологии, включая интерактивные формы обучения

Обучение происходит в форме лекционных и лабораторных занятий, а также самостоятельной работы студентов.

Теоретический материал излагается на лекциях. Причем конспект лекций, который остается у студента в результате прослушивания лекции не может заменить учебник. Его цель-формулировка основных утверждений и определений. Прослушав лекцию, полезно ознакомиться с более подробным изложением материала в учебнике. Список литературы разделен на две категории: необходимый для сдачи зачета минимум и дополнительная литература.

Изучение курса подразумевает не только овладение теоретическим материалом, но и получение практических навыков для более глубокого понимания разделов на основе решения задач и упражнений, иллюстрирующих доказываемые теоретические положения, а также развитие абстрактного мышления и способности самостоятельно доказывать утверждения.

Самостоятельная работа предполагает выполнение домашних работ. Практические задания, выполненные в аудитории, предназначены для указания общих методов решения задач определенного типа. Закрепить навыки можно лишь в результате самостоятельной работы. Кроме того, самостоятельная работа включает подготовку к зачету. При подготовке к сдаче зачета весь объем работы рекомендуется распределять равномерно по дням, отведенным для подготовки к зачету, контролировать каждый день выполнения работы. Лучше, если можно перевыполнить план. Тогда будет резерв времени.

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

### **Тема 1. Основные понятия конструирования ПО.**

Письменное домашнее задание , примерные вопросы:

Создание UML-диаграмм для планируемой программной системы

### **Тема 2. Проектирование при конструировании.**

Письменное домашнее задание , примерные вопросы:

Проектирование классов планируемой программной системы

### **Тема 3. Методология конструирования программного кода.**

Компьютерная программа , примерные вопросы:

Реализация взаимодействия классов. Формирование подсистем создаваемой программной системы.

### **Тема 4. Защитное программирование.**

Компьютерная программа , примерные вопросы:

Рефакторинг и улучшение обработки ошибок в создаваемой программной системе

### **Тема 5. Создание качественного кода.**

Отчет , примерные вопросы:

Оценка качества созданной системы

### **Тема 6. Тестирование в конструировании.**

Компьютерная программа , примерные вопросы:

Написание модульных тестов для созданной системы

### **Итоговая форма контроля**

зачет (в 6 семестре)

Примерные вопросы к итоговой форме контроля

1. Понятие программного обеспечения.
2. Жизненный цикл программного обеспечения.
3. Модели разработки программного обеспечения.
4. Объектно-ориентированный подход к разработке.
5. Конструирование программного обеспечения, его место в жизненном цикле.
6. Методологии конструирования программного обеспечения.
7. Проектирование программных систем, его место в жизненном цикле
8. Архитектурные стили разработки программных систем.
9. Принципы проектирования при конструировании.
10. Назначение проектирования при конструировании.
11. UML диаграмма классов как метод проектирования при конструировании.
12. ER-диаграмма как метод проектирования при конструировании.



13. Клиент-серверное программное обеспечение.
14. Понятие стандарта REST API.
15. Защитное программирование: обработка ошибок клиентского ввода.
16. Защитное программирование: механизм обработки исключений.
17. Тестирование ПО: виды тестирования и место в жизненном цикле.
18. Модульное тестирование при конструировании.
19. Разработка, управляемая тестированием.
20. Оценка качества ПО: метрики и инструменты.

### **7.1. Основная литература:**

1. Архитектура и проектирование программных систем: Монография / С.В. Назаров. - М.: НИЦ Инфра-М, 2013. - 351 с. Режим доступа: <http://znanium.com/catalog/product/353187>
2. Управление качеством программного обеспечения: Учебник / Б.В. Черников. - М.: ИД ФОРУМ: ИНФРА-М, 2012. - 240 с. Режим доступа: <http://znanium.com/catalog/product/256901>
3. Основы построения автоматизированных информационных систем: Учебник / В.А. Гвоздева, И.Ю. Лаврентьева. - М.: ИД ФОРУМ: НИЦ Инфра-М, 2013. - 320 с. Режим доступа: <http://znanium.com/catalog/product/392285>

### **7.2. Дополнительная литература:**

1. Технология разработки программного обеспечения: Учеб. пос. / Л.Г.Гагарина, Е.В.Кокорева, Б.Д.Виснадул; Под ред. проф. Л.Г.Гагариной - М.: ИД ФОРУМ: НИЦ Инфра-М, 2013. - 400 с. -Режим доступа: <http://znanium.com/catalog/product/389963>
2. Информационные технологии: разработка информационных моделей и систем: Учеб. пос. / А.В.Затонский - М.: ИЦ РИОР: НИЦ ИНФРА-М, 2014 - 344с. - Режим доступа: <http://znanium.com/catalog/product/400563>
3. Разработка и эксплуатация автоматизированных информационных систем: Учебное пособие / Гагарина Л.Г. М.:ИД ФОРУМ, НИЦ ИНФРА-М, 2013. - 384 с. - Режим доступа: <http://znanium.com/catalog/product/368454>

### **7.3. Интернет-ресурсы:**

- AlgoList - алгоритмы, методы, исходники - <http://algotlist.manual.ru/>  
<http://studme.org/> - <http://studme.org/>  
Systems Engineering Thinking Wiki - <http://sewiki.ru/>  
Интернет университет информационных технологий - <http://www.intuit.ru>  
Уроки по разработке веб-приложений на различных языках программирования - [metanit.com](http://metanit.com)

## **8. Материально-техническое обеспечение дисциплины(модуля)**

Освоение дисциплины "Конструирование программного обеспечения" предполагает использование следующего материально-технического обеспечения:

Компьютерный класс, представляющий собой рабочее место преподавателя и не менее 15 рабочих мест студентов, включающих компьютерный стол, стул, персональный компьютер, лицензионное программное обеспечение. Каждый компьютер имеет широкополосный доступ в сеть Интернет. Все компьютеры подключены к корпоративной компьютерной сети КФУ и находятся в едином домене.



Учебно-методическая литература для данной дисциплины имеется в наличии в электронно-библиотечной системе "ZNANIUM.COM", доступ к которой предоставлен студентам. ЭБС "ZNANIUM.COM" содержит произведения крупнейших российских учёных, руководителей государственных органов, преподавателей ведущих вузов страны, высококвалифицированных специалистов в различных сферах бизнеса. Фонд библиотеки сформирован с учетом всех изменений образовательных стандартов и включает учебники, учебные пособия, УМК, монографии, авторефераты, диссертации, энциклопедии, словари и справочники, законодательно-нормативные документы, специальные периодические издания и издания, выпускаемые издательствами вузов. В настоящее время ЭБС ZNANIUM.COM соответствует всем требованиям федеральных государственных образовательных стандартов высшего профессионального образования (ФГОС ВПО) нового поколения.

Так как дисциплина не предусматривает конкретизацию используемых программных средств разработки и языков программирования, то уведомление о необходимости их установки осуществляется во время учебного процесса исходя из выбора студентом языка реализации программного продукта.

Программа составлена в соответствии с требованиями ФГОС ВПО и учебным планом по направлению 09.03.04 "Программная инженерия" и профилю подготовки Технологии разработки информационных систем .

Автор(ы):

Прокопьев Н.А. \_\_\_\_\_

"\_\_" \_\_\_\_\_ 201\_\_ г.

Рецензент(ы):

Гусенков А.М. \_\_\_\_\_

"\_\_" \_\_\_\_\_ 201\_\_ г.