

МИНОБРНАУКИ РОССИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «ВГУ»)

**УТВЕРЖДАЮ**

Заведующий кафедрой  
программирования и информационных технологий



проф. Махортов С.Д.,  
*подпись, расшифровка подписи*  
03.05.2023 г.

**РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ**  
Б1.В.ДВ.04.02 Разработка программных приложений

**1. Код и наименование направления подготовки/специальности:**

09.03.03 Прикладная информатика

**2. Профиль подготовки/специализация/магистерская программа:**

"Прикладная информатика в экономике"

**3. Квалификация (степень) выпускника:** бакалавр

**4. Форма обучения:** очная

**5. Кафедра, отвечающая за реализацию дисциплины:**

программирования и информационных технологий

**6. Составители программы:** Вахтин Алексей Александрович, к.ф.-м.н., доц.

**7. Рекомендована:** НМС ф-та компьютерных наук, протокол № 7 от 03.05.2023 г.

**8. Учебный год:** 2023-2024

**Семестр(ы):** 4

**9. Цели и задачи учебной дисциплины:** Целью изучения дисциплины является формирование теоретических и практических навыков в области создания надежного и качественного программного обеспечения с применением современных технологий программирования, методов и средств коллективной разработки.

Основные задачи дисциплины:

- освоение теоретических основ и современных технологий анализа, проектирования и разработки программного обеспечения;
- овладение практическими навыками проектирования и разработки различных видов программного обеспечения на основе объектно-ориентированного подхода;
- приобретение опыта разработки программных средств средней сложности;
- знакомство с библиотеками классов и инструментальными средствами, используемыми при разработке программного обеспечения.

**10. Место учебной дисциплины в структуре ООП:** Дисциплина является курсом по выбору, относится к Блоку 1. Для ее изучения требуются входные знания курса «Программирование».

**11. Планируемые результаты обучения по дисциплине/модулю (знания, умения, навыки), соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями выпускников):**

Код	Название компетенции	Код(ы)	Индикатор(ы)	Планируемые результаты обучения
ПК-2	Способность составлять технико-экономическое обоснование проектных решений и техническое задание на разработку информационной системы	ПК-2.1	Разработка требований и проектирование программного обеспечения	Знать: технологии разработки программных приложений  Уметь: разрабатывать программные приложения  Владеть: технологией разработки программных приложений
		ПК-2.2	Разработка требований и проектирование технического обеспечения	Уметь: определять техническое обеспечение для программных приложений
ПК-3	Способность разрабатывать прикладное программное обеспечение и создавать прототипы информационных систем	ПК-3.1	Разработка прототипов ИС на базе типовой ИС	Знать: назначение и последовательность этапов разработки программных средств. Стандарты в области индустрии программного обеспечения. Критерии качества программного обеспечения и способы его оценивания  Уметь: выполнять отдельные этапы процесса разработки программного обеспечения в соответствии с требованиями стандартов  Владеть (иметь навык(и)): основными инструментальными средствами разработки программного обеспечения
		ПК-3.2	Кодирование на языках программирования	Знать: назначение и последовательность этапов разработки программных средств. Стандарты в области индустрии

				<p>программного обеспечения. Критерии качества программного обеспечения и способы его оценивания</p> <p>Уметь: выполнять отдельные этапы процесса разработки программного обеспечения в соответствии с требованиями стандартов</p> <p>Владеть (иметь навык(и)): основными инструментальными средствами разработки программного обеспечения</p>
ПК-6	Способность документировать процессы создания информационных систем на стадиях жизненного цикла	ПК-6.1	Создание пользовательской документации к ИС	<p>Знать: технологии создания пользовательской документации для программного приложения</p> <p>Уметь: создавать пользовательскую документацию для программного приложения</p> <p>Владеть (иметь навык(и)): технологиями создания пользовательской документации для программного приложения</p>
		ПК-6.2	Методологическое обеспечение обучения пользователей ИС	<p>Знать: методологическое обеспечение обучения пользователей программного приложения</p> <p>Уметь: применять методологическое обеспечение обучения пользователей программного приложения</p> <p>владеть (иметь навык(и)): методологическим обеспечением обучения пользователей программного приложения</p>

**12. Объем дисциплины в зачетных единицах/час. — 3/108.**

**Форма промежуточной аттестации – зачет с оценкой.**

### 13. Виды учебной работы

Вид учебной работы	Трудоемкость			
	Всего	По семестрам		
		3 семестр		
Аудиторные занятия	64	64		
в том числе: лекции	16	16		
практические	32	32		
лабораторные	16	16		
Самостоятельная работа	44	44		
Форма промежуточной аттестации (зачет – 0 час. / экзамен – __ час.)	0	0		
Итого:	108	108		

### 13.1. Содержание дисциплины

п/п	Наименование раздела дисциплины	Содержание раздела дисциплины
1.1	Основные понятия технологии программирования	Индустриальный подход к разработке программного обеспечения. Методы и средства программной инженерии. Жизненный цикл программного продукта. Этапы процесса разработки. Понятие качества программного продукта, основные критерии качества
1.2	Модели процесса разработки программных средств (ПС): прогностические и адаптивные модели	Стратегии разработки и модели процесса разработки. Прогностические и адаптивные модели. Особенности прогностических моделей. Каскадная, инкрементная и спиральная модели процесса разработки ПС. RUP-процесс. Особенности адаптивных моделей. XP-модель и принципы экстремального программирования. Scrum-модель.
1.3	Моделирование предметной области; структурный и объектно-ориентированный анализ	Анализ и моделирование предметной области как основа для разработки требований к ПО. Метод функционального моделирования SADT. Состав функциональной модели. IDEF- и DFD-диаграммы. Принципы и средства объектного моделирования систем. Унифицированный язык моделирования UML. Виды диаграмм. Концептуальный и логический уровни моделирования.
1.4	Проектирование программных средств; CASE-средства архитектурного и детального проектирования	Архитектурное и детальное проектирование. Основные виды архитектур программных систем. Задачи этапа архитектурного проектирования: выявление подсистем и интерфейсов, формирование архитектурных уровней, проектирование структуры потоков управления. Уровень логического (детального) проектирования.
1.5	Особенности объектно-ориентированного проектирования программных систем. Паттерны проектирования.	Проектирование объектно-ориентированных ПС. Проектирование классов и интерфейсов. Шаблоны проектирования. Объектно-ориентированное CASE средство Rational Rose.
1.6	Современные языки программирования: особенности и тенденции развития. Техники написания эффективного программного кода.	Классификация языков программирования: процедурные, объектно-ориентированные и декларативные. Критерии сравнительного анализа языков. Проблемы совместимости компонент, написанных на различных языках программирования.
1.7	Тестирование и отладка программных средств; виды тестирования	Тестирование и отладка программных средств. Виды тестирования. Тестовые наборы и тестовые процедуры. Технологии разработки, вводимые тестированием. Автоматизация процесса тестирования модулей. Инструментальное средство NUnit.
1.8	Проблема контроля версий в процессе разработки ПС	Понятие версии ПС и контроля версий. Автоматизация контроля версий. Утилита Subversion.
1.9	Стандартизация в сфере программной инженерии	Стандартизация в сфере программной инженерии. Национальные и международные стандарты. Стандарты группы ISO/IEC: стандарты на базовые процессы и стандарты оценки уровня зрелости.
1.10	Управление программным проектом	Цели и задачи управления проектом. Планирование проектных задач и распределения работ. Риски, анализ и управление рисками. LOC- и FP-метрики. Оценка проекта на основе метрик.

### 13.2. Темы (разделы) дисциплины и виды занятий

№ п/п	Наименование темы (раздела) дисциплины	Виды занятий (часов)				
		Лекции	Практические	Лабораторные	Самостоятельная работа	Всего
1	Основные понятия технологии программирования	1	2	1	4	8

2	Модели процесса разработки программных средств (ПС): прогностические и адаптивные модели	2	4	2	4	12
3	Моделирование предметной области; структурный и объектно-ориентированный анализ	2	4	2	4	12
4	Проектирование программных средств; CASE-средства архитектурного и детального проектирования	2	4	2	4	12
5	Особенности объектно-ориентированного проектирования программных систем. Паттерны проектирования.	2	4	2	4	12
6	Современные языки программирования: особенности и тенденции развития. Техники написания эффективного программного кода.	1	4	2	4	11
7	Тестирование и отладка программных средств; виды тестирования	2	4	2	4	12
8	Проблема контроля версий в процессе разработки ПС	1	2	2	6	11
9	Стандартизация в сфере программной инженерии	2	2		4	8
10	Управление программным проектом	1	2	1	6	10
	Итого:	16	32	16	44	108

#### 14. Методические указания для обучающихся по освоению дисциплины

Рекомендуется работа с конспектами лекций, презентационным материалом, выполнение практических проектов, тестов, заданий текущей аттестации. Учебные и методические материалы по дисциплине представлены на сетевом диске внутренней сети факультета.

#### 15. Перечень основной и дополнительной литературы, ресурсов интернет, необходимых для освоения дисциплины

а) основная литература:

№ п/п	Источник
1.	Орлов С.А. Технологии разработки программного обеспечения. Разработка сложных программных систем: учебник для вузов /С.А. Орлов. – СПб.: Питер, 2012. – 527 с.

б) дополнительная литература:

№ п/п	Источник
2.	Кулямин В.В. Технологии программирования. Компонентный подход: учебное пособие/В.В. Кулямин. – М. Бином. Лаборатория знаний. 2007. – 463 с.

в) информационные электронно-образовательные ресурсы (официальные ресурсы интернет)\*:

№ п/п	Ресурс
3.	В.А. Петрухин, Е.М. Лаврищева Методы и средства инженерии программного обеспечения. <a href="http://www.intuit.ru/department/se/swebok/0/">http://www.intuit.ru/department/se/swebok/0/</a>
4.	<a href="http://www.lib.vsu.ru">www.lib.vsu.ru</a> ЗНБ ВГУ

## 16. Перечень учебно-методического обеспечения для самостоятельной работы

№ п/п	Источник
1.	Контрольные задания и тесты в электронном варианте
2.	Сетевой диск внутренней сети факультета

## 17. Информационные технологии, используемые для реализации учебной дисциплины, включая программное обеспечение и информационно-справочные системы (при необходимости)

## 18. Материально-техническое обеспечение дисциплины:

Мультимедийная лекционная аудитория, компьютерные классы факультета компьютерных наук.

## 19. Оценочные средства для проведения текущей и промежуточной аттестаций

Порядок оценки освоения обучающимися учебного материала определяется содержанием следующих разделов дисциплины:

№ п/п	Наименование раздела дисциплины (модуля)	Компетенция(и)	Индикатор(ы) достижения компетенции	Оценочные средства
1	Основные понятия технологии программирования	ПК-2 ПК-3 ПК-6	ПК-2.1, ПК-2.2 ПК-3.1, ПК-3.2 ПК-6.1, ПК-6.2	практический проект, лабораторные работы
2	Модели процесса разработки программных средств (ПС): прогностические и адаптивные модели	ПК-2 ПК-3 ПК-6	ПК-2.1, ПК-2.2 ПК-3.1, ПК-3.2 ПК-6.1, ПК-6.2	практический проект, лабораторные работы
3	Моделирование предметной области; структурный и объектно-ориентированный анализ	ПК-2 ПК-3 ПК-6	ПК-2.1, ПК-2.2 ПК-3.1, ПК-3.2 ПК-6.1, ПК-6.2	практический проект, лабораторные работы
4	Проектирование программных средств; CASE-средства архитектурного и детального проектирования	ПК-2 ПК-3 ПК-6	ПК-2.1, ПК-2.2 ПК-3.1, ПК-3.2 ПК-6.1, ПК-6.2	практический проект, лабораторные работы
5	Особенности объектно-ориентированного проектирования программных систем. Паттерны проектирования.	ПК-2 ПК-3 ПК-6	ПК-2.1, ПК-2.2 ПК-3.1, ПК-3.2 ПК-6.1, ПК-6.2	практический проект, лабораторные работы
6	Современные языки программирования: особенности и тенденции развития. Техники написания эффективного программного кода.	ПК-2 ПК-3 ПК-6	ПК-2.1, ПК-2.2 ПК-3.1, ПК-3.2 ПК-6.1, ПК-6.2	практический проект, лабораторные работы
7	Тестирование и отладка программных средств; виды тестирования	ПК-2 ПК-3 ПК-6	ПК-2.1, ПК-2.2 ПК-3.1, ПК-3.2 ПК-6.1, ПК-6.2	практический проект, лабораторные работы
8	Проблема контроля версий в процессе разработки ПС	ПК-2 ПК-3 ПК-6	ПК-2.1, ПК-2.2 ПК-3.1, ПК-3.2 ПК-6.1, ПК-6.2	практический проект, лабораторные работы
9	Стандартизация в сфере программной инженерии	ПК-2 ПК-3 ПК-6	ПК-2.1, ПК-2.2 ПК-3.1, ПК-3.2 ПК-6.1, ПК-6.2	практический проект, лабораторные работы
10	Управление программным проектом	ПК-2 ПК-3 ПК-6	ПК-2.1, ПК-2.2 ПК-3.1, ПК-3.2 ПК-6.1, ПК-6.2	практический проект, лабораторные работы
Промежуточная аттестация форма контроля – зачет				КИМ, практический проект, лабораторная работа

## 20 Типовые оценочные средства и методические материалы, определяющие процедуры оценивания

### 20.1 Текущий контроль успеваемости

Контроль успеваемости по дисциплине осуществляется с помощью контроля и оценки следующих практических заданий: проектирование, реализация, тестирование и приложения для предметной области по выбору обучающегося, а также подготовка и защита курсовой работы.

### 20.2 Промежуточная аттестация

Промежуточная аттестация по дисциплине осуществляется по итогам результатов решений поставленных задач на практике и лабораторных работах. Перечень заданий приведен выше.

Промежуточная аттестация проводится в соответствии с Положением о промежуточной аттестации обучающихся по программам высшего образования. При оценивании используются количественные шкалы оценок.

Соотношение показателей, критериев и шкалы оценивания результатов обучения.

Критерии оценивания компетенций	Уровень сформированности компетенций	Шкала оценок
Обучающийся в полной мере владеет теоретическими основами дисциплины, способен иллюстрировать ответ примерами, применять теоретические знания для решения практических задач.	<i>Повышенный уровень</i>	<i>отлично</i>
Ответ на контрольно-измерительный материал не вполне соответствует двум из перечисленных показателей, но обучающийся дает правильные ответы на дополнительные вопросы. Недостаточно продемонстрировано владение системным подходом, или содержатся отдельные пробелы при описании терминологии дисциплины и ее практик.	<i>Базовый уровень</i>	<i>хорошо</i>
Ответ на контрольно-измерительный материал не соответствует любым трем из перечисленных показателей, обучающийся дает неполные ответы на дополнительные вопросы. Демонстрирует лишь частичные знания, или не умеет связывать теорию с практикой, или имеет неполное представление о системном подходе, допускает существенные ошибки.	<i>Пороговый уровень</i>	<i>удовлетворительно</i>
Ответ на контрольно-измерительный материал не соответствует любым четырем из перечисленных показателей. Обучающийся демонстрирует отрывочные, фрагментарные знания, допускает грубые ошибки.	–	<i>Неудовлетворительно</i>

#### 20.2.1 Перечень вопросов к зачету:

1	<i>Что такое технология программирования? Методы и средства разработки программных продуктов?</i>
2	<i>Понятие качества программных продуктов. Критерии качества.</i>
3	<i>Определение качества ПО в стандарте ISO 9126. Аспекты качества, их взаимное влияние.</i>
4	<i>Многоуровневая модель качества ПО в стандарте ISO 9126.</i>
5	<i>Модель жизненного цикла программных средств. Фазы жизненного цикла. Этапы фазы разработки, их содержание.</i>
6	<i>Анализ предметной области: цели и задачи. Модели предметной области. Формальные определения. Классификация моделей.</i>
7	<i>Методология IDEF0, синтаксис IDEF0-моделей.</i>
8	<i>Диаграммы потоков данных (DFD-диаграммы), их использование при моделировании предметной области.</i>
9	<i>Модели процесса разработки программных средств и реализуемые ими стратегии. Прогностические модели (процессы) разработки. Каскадная модель, ее характеристика.</i>
10	<i>Модели процесса разработки программных средств и реализуемые ими стратегии. Прогностические модели (процессы) разработки. Инкрементная модель процесса разработки, ее характеристика.</i>

11	<i>Модели процесса разработки программных средств и реализуемые ими стратегии. Прогностические модели (процессы) разработки. Спиральная модель процесса разработки, ее характеристика.</i>
12	<i>RUP-модель процесса разработки программного средства: принципы и структура.</i>
13	<i>Адаптивные процессы разработки программных средств. XP-модель процесса разработки. Принципы экстремального программирования.</i>
14	<i>Адаптивные процессы разработки программных средств. Scrum-модель процесса разработки.</i>
15	<i>Задачи этапа объектно-ориентированного анализа предметной области. Методика определения границ системы и ключевых абстракций. Пример проведения анализа.</i>
16	<i>Функциональные требования к системе. Способ их представления в виде UML-диаграммы. Пример диаграммы с использованием отношений «расширяет» и «включает».</i>
17	<i>Понятие прецедента и сценария. Пример прецедента, основного и дополнительного сценариев.</i>
18	<i>Нефункциональные требования к системе, их виды. Примеры нефункциональных требований.</i>
19	<i>Концептуальная модель системы: концептуальные классы, системные события и системные операции. Способ их представления в виде UML-диаграмм. Пример концептуального описания прецедента.</i>
20	<i>Диаграммы взаимодействия как элементы концептуальной модели. Синтаксис диаграмм взаимодействия. Примеры диаграмм взаимодействия</i>
21	<i>Цели и задачи этапа проектирования. Понятие модели проектирования, ее отличия от концептуальной модели.</i>
22	<i>Стадии проектирования, их краткая характеристика.</i>
23	<i>Задачи, решаемые на стадии эскизного проектирования.</i>
24	<i>Понятие архитектуры ПС. Проблема выбора архитектуры. Влияние архитектуры на качественные характеристики ПС.</i>
25	<i>Понятие модуля и модульного программирования. Преимущества модульного подхода к разработке ПО.</i>
26	<i>Задачи, решаемые на стадии детального проектирования.</i>
27	<i>Цели и задачи проектирования пользовательского интерфейса.</i>
28	<i>Особенности объектно-ориентированного проектирования.</i>
29	<i>Идентификация методов программных классов. Диаграммы классов, способы отображения отношений ассоциации и зависимости. Пример диаграммы классов.</i>
30	<i>Обязанности программных классов, виды обязанностей. Визуализация распределения обязанностей посредством UML-диаграмм. Примеры диаграмм.</i>
31	<i>Шаблоны проектирования, их классификация. Правила описания шаблонов, примеры шаблонов с их описаниями.</i>
32	<i>Шаблоны анализа (analysis patterns), их классификация.</i>
33	<i>Архитектурные шаблоны (architectural patterns), их классификация.</i>
34	<i>Шаблоны проектирования (design patterns), их примеры.</i>
35	<i>Шаблоны распределения обязанностей, их назначение. Примеры применения.</i>
36	<i>Структурные шаблоны, их назначение. Примеры структурных шаблонов с их описаниями.</i>
37	<i>Тестирование программного средства. Стадии тестирования и их характеристика.</i>
39	<i>Основные принципы тестирования.</i>
40	<i>Тесты и тестовые наборы. Понятие тестового покрытия. Отладочное тестирование. Соотношение структурного и функционального подходов.</i>
41	<i>Структурный подход к формированию тестовых наборов. Пример реализации структурного подхода.</i>
42	<i>Функциональный подход к формированию тестовых наборов. Пример реализации функционального подхода.</i>
43	<i>Интеграционное тестирование. Виды интеграционного тестирования. Критерии полноты тестовых наборов.</i>
44	<i>Регрессионное тестирование. Критерии завершения отладочного тестирования.</i>
45	<i>Восходящая и нисходящая стратегии интеграционного тестирования, механизмы их реализации.</i>
46	<i>Системное тестирование. Виды системного тестирования. Критерии полноты тестовых наборов.</i>
47	<i>Особенности объектно-ориентированного тестирования. Расширение области применения тестирования. Критерии тестирования моделей.</i>
48	<i>Особенности методики модульного тестирования объектно-ориентированных систем. Тестирование классов.</i>
49	<i>Особенности методики интеграционного тестирования объектно-ориентированных систем. Тестирование кластеров и потоковое тестирование.</i>
50	<i>Понятие автоматизированного тестирования. Автотесты. Достоинства и недостатки автоматизированного тестирования.</i>
51	<i>Типы автоматизированного тестирования, их цели. Средства автоматизированного тестирования.</i>



52	<i>Утверждения, параметры утверждений. Группы утверждений, классическая и закрытая модель утверждений.</i>
56	<i>Система конкурирующих версий CVS, ее достоинства и недостатки.</i>
57	<i>Система Subversion, ее архитектура; достоинства и недостатки системы.</i>
58	<i>Хранилище, его структура, правки. Команды SVN для работы с хранилищем.</i>
59	<i>Понятия рабочей копии и служебного каталога. Команды SVN для работы с рабочими копиями.</i>
60	<i>Сценарий объединения правок. Конфликты и способы их разрешения.</i>
61	<i>Понятие сборки, манифест сборки. Сборка приложения, системы автоматизации сборки.</i>
62	<i>Утилита NAnt, файл сборки и его структура.</i>
63	<i>Цели, зависимость целей, описание целей.</i>
64	<i>Документирование процесса разработки. Типы документов управления.</i>
65	<i>Документирование программного продукта. Документация сопровождения, ее назначение и состав.</i>
66	<i>Документирование программного продукта. Пользовательская документация, ее назначение и состав.</i>
67	<i>Генератор документации Sandcastle, его назначение и принцип работы.</i>
68	<i>Руководство проектом и особенности проектной деятельности.</i>
69	<i>Проектная команда, группы и роли в проектной команде.</i>
70	<i>Критерии оценивания проектов, шкалы ценности проекта.</i>
71	<i>Риски, их ранжирование, управление рисками.</i>
72	<i>Базовое расписание проекта, точки контроля, распараллеливание работ.</i>
73	<i>Способы контроля хода выполнения проекта: меры и метрики. Виды метрик</i>