

**МИНОБРНАУКИ РОССИИ**  
**федеральное государственное бюджетное образовательное учреждение**  
**высшего образования**  
**Ярославский государственный университет им. П.Г.Демидова**

**УЧЕБНО-МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ**  
**ПО ДИСЦИПЛИНЕ**

**Современные средства анализа данных**

Направление подготовки (специальность):  
09.04.03 ПРИКЛАДНАЯ ИНФОРМАТИКА

Образовательная программа  
Искусственный интеллект в корпоративных информационных системах

**очная форма обучения**

Составитель:  
Краснов М.В., к.ф.-м.н., доцент

г. Ярославль

## **Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины (модуля)**

Основная литература:

Основная литература:

1. Приемы объектно-ориентированного проектирования : паттерны проектирования : [принципы применения паттернов проектирования, классификация паттернов, различные подходы к выбору паттернов, каталог паттернов с детальным их описанием] / Э. Гамма, Р. Хелм, СПб., Питер, 2013, 366с.

Дополнительная литература:

1. С. Макконнелл. Совершенный код. Мастер-класс. М.: Русская редакция. - СПб.: Питер, 2012.
2. Уткин, В. Б., Информационные технологии управления : учебник для вузов / В. Б. Уткин, К. В. Балдин, М., Академия, 2008, 395с.

## **Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины (модуля)**

1. Электронная библиотека учебных материалов ЯрГУ ([http://www.lib.uni Yar.ac.ru/opac/bk\\_cat\\_find.php](http://www.lib.uni Yar.ac.ru/opac/bk_cat_find.php)).
2. Информационная система "Единое окно доступа к образовательным ресурсам" (<http://www.edu.ru> (раздел Учебно-методическая библиотека) или по прямой ссылке <http://window.edu.ru/library>).
3. Электронно-библиотечная система «Университетская библиотека online» ([www.biblioclub.ru](http://www.biblioclub.ru) ).

Учебно-методические указания и рекомендации  
к изучению тем лекционных и практических занятий, самостоятельной  
работе студентов

## Очная форма обучения

### Лекции

№	Тема лекции	Количество часов
1	Система стандартов POSIX.	2
2	Организация процесса программирования, стратегии планирования. Гибкие методологии разработки программ (Agile, Extreme Programming), планирование итераций.	1
3	Жизненный цикл программного проекта. Стили кодирования. Управление исходным кодом, системы контроля версий. Разработка документации. Системы автоматического построения документации.	1
4	Технологии тестирования. Реверсивное тестирование, приемо-сдаточное тестирование, модульные тесты. Разработка на основе тестирования.	1
5	Паттерны проектирования.	1
6	Система стандартов POSIX.	2
	<b>Итого:</b>	8

### Практические занятия

№	Тема практического занятия	Количество часов
1	Система стандартов POSIX.	5
2	Организация процесса программирования, стратегии планирования. Гибкие методологии разработки программ (Agile, Extreme Programming), планирование итераций.	4
3	Жизненный цикл программного проекта. Стили кодирования. Управление исходным кодом, системы контроля версий. Разработка документации. Системы автоматического построения документации.	3
4	Технологии тестирования. Реверсивное тестирование, приемо-сдаточное тестирование, модульные тесты. Разработка на основе тестирования.	3
5	Паттерны проектирования.	4
6	Система стандартов POSIX.	3
	<b>Итого:</b>	22

### Типовой вариант контрольной работы

Студенту выдается билет с двумя вопросами по одному вопросу из каждой части.

#### Часть 1.

1. Стандарт POXIX: структура файловой системы.
2. Стандарт POXIX: командная оболочка.
3. Стандарт POXIX: стандартная библиотека.
4. IPC и сокеты в стандарте POSIX.
5. Гибкие методологии разработки ПО. Основные принципы.
6. Гибкая методология разработки Scrum.
7. Принципы экстремального программирования.
8. Стандарты, процессы, стадии жизненного цикла ПО.
9. Водопадная модель процесса разработки ПО.
10. Итерационная модель жизненного цикла ПО.
11. Спиральная модель жизненного цикла ПО.
12. Системы контроля версий, отличие централизованных и распределенных систем.
13. Типы документации, User manual.
14. Документирование исходного кода, системы автоматического построения документации. Пример использования Doxygen.
15. Классификация видов тестирования, уровни тестирования, понятие тестового покрытия. Unit тесты.
16. Реверсивное, приемо-сдаточное и модульное тестирование.
17. Разработка программного обеспечения на основе тестирования.

#### Часть 2. Паттерны проектирования

1. Паттерн [Abstract Factory](#).
2. Паттерн [Adapter](#).
3. Паттерн [Bridge](#).
4. Паттерн [Builder](#).
5. Паттерн [Chain of Responsibility](#).
6. Паттерн [Command](#).
7. Паттерн [Composite](#).
8. Паттерн [Decorator](#).
9. Паттерн [Facade](#).
10. Паттерн [Factory Method](#).
11. Паттерн [Flyweight](#).
12. Паттерн [Interpreter](#).
13. Паттерн [Iterator](#).
14. Паттерн [Mediator](#).
15. Паттерн [Memento](#).
16. Паттерн [Object Pool](#).
17. Паттерн [Observer](#).

18. Паттерн [Prototype](#).
19. Паттерн [Proxy](#).
20. Паттерн [Singleton](#).
21. Паттерн [State](#). Паттерн [Strategy](#).
22. Паттерн [Template Method](#).
23. Паттерн [Visitor](#).

### **Темы рефератов по курсу “Анализ Информационных Технологий”**

1. Классификация языков программирования, принципиальные отличия.
2. Профиль систем реального времени.
3. Big Data. Что такое Spark, Hadoop.
4. Система стандартов OSI.
5. Безопасность сервисов операционных систем.
6. Качество ПО. Показатели качества.
7. Статический анализ кода.
8. Сетевые атаки. Виды. Способы борьбы.
9. Модульные тесты. Разработка через тестирование (TDD).
10. Вирусы. Классификация, симптомы, лечение.
11. Что такое Docker, как и зачем применять. Основные понятия, плюсы, минусы.
12. Создание Docker машины (практическое руководство).
13. Облачные вычисления.
14. Инструменты машинного обучения.
15. Сравнительный анализ Javascript Framework.
16. Микросервисы. Особенности. Инструменты для реализации.

### **Список заданий к зачету**

Зачет выставляется по результатам контрольной работы и подготовленного реферата.