

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

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

*ТЕХНОЛОГИИ ФУНКЦИОНАЛЬНОГО ПРОГРАММИРОВАНИЯ В  
СОВРЕМЕННЫХ ИНФОРМАЦИОННЫХ СИСТЕМАХ*

Направление подготовки (специальность):

02.04.02 Фундаментальная информатика и информационные технологии

Образовательная программа

Искусственный интеллект и компьютерные науки

**очная форма обучения**

Составитель:

**ВАСИЛЬЕВ АНДРЕЙ МИХАЙЛОВИЧ,**  
**ДОЦЕНТ Ф-ТА ИВТЯРГУ ИМ. П.Г. ДЕМИДОВА**

г. Ярославль

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

Основная литература:

1 Салмина, Н. Ю. Функциональное программирование и интеллектуальные системы : учебное пособие / Н. Ю. Салмина. — Томск : Томский государственный университет систем управления и радиоэлектроники, 2016. — 100 с.

Дополнительная литература:

1 Кубенский А. А. Функциональное программирование. – 2018.. URL: <https://elibrary.ru/item.asp?id=37484990>

2 Душкин Р. В. Функциональное программирование на языке Haskell. Учебное пособие. – 2008. URL: <https://elibrary.ru/item.asp?id=20242406>

### **Учебно-методическое обеспечение самостоятельной работы обучающихся по дисциплине (модулю)**

1 Электронный университет Moodle ЯрГУ URL: <https://moodle.uniyar.ac.ru/>

2 Единое окно доступа к образовательным ресурсам URL: <http://window.edu.ru/>

### **Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины (модуля), включая перечень информационных справочных систем (при необходимости)**

1 Библиотека lodash, реализующая методы функционального программирования в языке JavaScript. URL: <https://lodash.com>

2 Учебные материалы по языку программирования JavaScript на ресурсе Mozilla Developer Network. URL: <https://developer.mozilla.org/ru/docs/Web/JavaScript>

### **Перечень информационных технологий, используемых при изучении дисциплины, включая программное обеспечение**

1. Интерпретатор языка JavaScript в рамках веб-браузера Firefox (свободно распространяемое ПО)
2. Среда Visual Studio Code (свободно распространяемое ПО)

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

**Содержание дисциплины**

Наименование раздела дисциплины	Название темы с кратким содержанием
1.	Введение. Функциональные языки программирования (ФЯП). Области применения ФЯП, взаимодействие с другими парадигмами программирования.
2.	Категоризация исходного кода программного обеспечения (ПО): данные, вычисления и действия. Методы анализа кода с целью категоризации, в том числе с целью повышения надёжности работы приложений.
3.	Методы рефакторинга исходного кода ПО для уменьшения количества действий. Выделение вычислений из действий, отказ от неявных входов и выходов в действиях.
4.	Методы обеспечения неизменяемости данных ПО: копирование при записи и глубокое копирование. Организация взаимодействия между изменяемыми и неизменяемыми данными внутри ПО.
5.	Подходы к проектированию архитектуры приложений согласно стратифицированному подходу. Шаблоны несложной реализации, барьера из абстракций, минимального интерфейса и удобных слоёв.
6.	Функции как объекты первого рода. Функциональное итерирование. Использование итераторов filter, map, reduce. Формирование цепочек по обработке данных.

**Задания для выполнения рефератов**

№	Тема	Описание темы	Источники
1	Анализ средств применения функциональных инструментов в языке программирования Java	Выяснение списка доступных функциональных инструментов, предоставляемых средствами разработки и доступными библиотеками. Анализ применимости рассмотренных методик по рефакторингу приложений.	<ul style="list-style-type: none"> <li>● <a href="https://docs.oracle.com/en/java/javase/17/">https://docs.oracle.com/en/java/javase/17/</a></li> <li>● <a href="http://www.functionaljava.org/">http://www.functionaljava.org/</a></li> <li>● <a href="https://code.google.com/archive/p/lambdaj/">https://code.google.com/archive/p/lambdaj/</a></li> </ul>
2	Анализ средств применения функциональных инструментов в языке	Выяснение списка доступных функциональных инструментов, предоставляемых средствами	<ul style="list-style-type: none"> <li>● <a href="https://docs.python.org/3/">https://docs.python.org/3/</a></li> </ul>

	программирования Python	разработки и доступными библиотечками. Анализ применимости рассмотренных методик по рефакторингу приложений.	<ul style="list-style-type: none"> <li>• <a href="https://fancy.readthedocs.io/en/stable/">https://fancy.readthedocs.io/en/stable/</a></li> <li>• <a href="https://toolz.readthedocs.io/en/latest/">https://toolz.readthedocs.io/en/latest/</a></li> </ul>
3	Анализ средств применения функциональных инструментов в языке программирования Kotlin	Выяснение списка доступных функциональных инструментов, предоставляемых средствами разработки и доступными библиотечками. Анализ применимости рассмотренных методик по рефакторингу приложений.	<a href="https://kotlinlang.org/docs/home.html">https://kotlinlang.org/docs/home.html</a>
4	Анализ средств применения функциональных инструментов в языке программирования Ruby	Выяснение списка доступных функциональных инструментов, предоставляемых средствами разработки и доступными библиотечками. Анализ применимости рассмотренных методик по рефакторингу приложений.	<ul style="list-style-type: none"> <li>• <a href="https://ruby-doc.org/">https://ruby-doc.org/</a></li> <li>• <a href="https://womanonrails.com/functional-programming-ruby">https://womanonrails.com/functional-programming-ruby</a></li> </ul>
5	Анализ средств применения функциональных инструментов в языке программирования Haskell	Выяснение списка доступных функциональных инструментов, предоставляемых средствами разработки и доступными библиотечками. Анализ способов взаимодействия с библиотеками на языке программирования Си.	<a href="https://www.haskell.org/documentation/">https://www.haskell.org/documentation/</a>
6	Анализ средств применения функциональных инструментов в языке программирования Erlang	Выяснение списка доступных функциональных инструментов, предоставляемых средствами разработки и доступными библиотечками. Анализ способов взаимодействия с библиотеками на языке программирования Си.	<a href="https://www.erlang.org/docs">https://www.erlang.org/docs</a>
7	Анализ средств применения функциональных инструментов в языке программирования Clojure	Выяснение списка доступных функциональных инструментов, предоставляемых средствами разработки и доступными библиотечками. Анализ способов взаимодействия с библиотеками на языке программирования Си.	<a href="https://clojure.org/guides/spec">https://clojure.org/guides/spec</a>
8	Анализ средств применения функциональных инструментов в языке программирования Elm	Выяснение списка доступных функциональных инструментов, предоставляемых средствами разработки и доступными библиотечками. Анализ способов взаимодействия с библиотеками на языке программирования Си.	<a href="https://elm-lang.org/docs">https://elm-lang.org/docs</a>

### Требования к оформлению и защите рефератов.

Реферат оформляется в электронном виде как файл формата pdf, защита осуществляется в виде доклада, сопровождаемого презентацией. Правила оформления реферата <https://kursach37.com/oformlenie-referata-po-gost/>.

## Критерии оценки

«Отлично» – оцениваются рефераты, содержание которых основано на глубоком и всестороннем знании темы, изученной литературы, изложено логично, аргументировано и в полном объеме. Основные понятия, выводы и обобщения сформулированы убедительно и доказательно.

«Хорошо» – оцениваются рефераты, основанные на твердом знании исследуемой темы. Возможны недостатки в систематизации или в обобщении материала, неточности в выводах. Студент твердо знает основные категории, умело применяет их для изложения материала.

«Удовлетворительно» – оцениваются рефераты, которые базируются на знании основ предмета, но имеются значительные пробелы в изложении материала, затруднения в его изложении и систематизации, выводы слабо аргументированы, в содержании допущены теоретические ошибки.

«Неудовлетворительно» – оцениваются рефераты, в которых обнаружено неверное изложение основных вопросов темы, обобщений и выводов нет. Текст реферата целиком или в значительной части дословно переписан из первоисточника без ссылок на него.

## Пример задания для выполнения лабораторных работ

### Требования к исходному коду

В качестве основы для применения техник рекомендуется использовать исходный код, который был разработан студентом самостоятельно. Подходит код любой лабораторной работы, выполненный для защиты результатов по предыдущим курсам по программированию. Необходимо использовать код, которым студенты смогут поделиться с другими слушателями курса.

Удобнее всего применять предложенные подходы к языкам программирования, ориентированных на запуск в средах с функцией автоматической сборки мусора. То есть данные подходы плохо применимы к системным языкам программирования (Си, C++, Rust).

Для работы рекомендуется выбирать код длиной не более двухсот строк. Можно взять подсистему приложения.

### Задание на доработку исходного кода

Необходимо применить методики рефакторинга исходного кода и подготовить выступление, в рамках которого рассмотреть процесс изменения исходного кода. Выступление должно включать описание следующих этапов модификации:

- Изначальное состояние исходного кода. В описание должно входить:
  - Описание назначение кода.
  - Описание изначальной связи методов друг с другом.
- Описание результата применения техник к коду. В описании должно быть:

- Пример применения техник к ряду методов.
- Изменение в структуре методов.

Помимо презентации также необходимо подготовить Git-репозиторий, который будет включать в себя как исходное состояние, так и фиксацию состояний во время применения техник рефакторинга исходного кода.

Подготовленный набор материалов: презентация и ссылка на репозиторий загружаются в качестве ответа на задание.

### Критерии оценки

«Отлично» – проанализирован и классифицирован весь исходный код, больше 50% исходного кода исправлено с применением задач рефакторинга.

«Хорошо» – проанализировано и классифицировано не менее 80% исходного код, больше 35% исходного кода исправлено с применением задач рефакторинга.

«Удовлетворительно» – проанализировано и классифицировано не менее 60% исходного код, больше 20% исходного кода исправлено с применением задач рефакторинга.

«Неудовлетворительно» – проанализировано и классифицировано менее 50% исходного кода, менее 20% исходного кода исправлено с применением задач рефакторинга.

### Вопросы к зачету

1. Перечислите области применения функциональных языков программирования.
2. Укажите сферы, в которых применение функциональных языков программирования затруднено. Объясните эти причины.
3. Разъясните категоризацию кода на данные, вычисления и действия.
4. Укажите ключевые характеристики категории исходного кода данные. Какие есть сильные стороны и ограничения у данной категории?
5. Укажите ключевые характеристики категории исходного кода вычисления. Какие есть сильные стороны и ограничения у данной категории?
6. Укажите ключевые характеристики категории исходного кода действия. Какие есть сильные стороны и ограничения у данной категории?
7. Опишите типичные рефакторинги, которые можно применить для выделения вычислений из действий. В каких случаях стоит использовать данные методы?
8. Следует ли применять данные методы к действиям, даже если невозможно из них выделить вычисления? Поясните на примерах.
9. Объясните шаблон копирования данных при записи. Для чего он может быть использован? Приведите пример использования.
10. Объясните шаблон глубокого копирования данных. Для чего он может быть использован? Приведите пример использования.
11. Какие из шаблонов копирования можно использовать для организации данных приложения в неизменяемом формате? Укажите их достоинства и недостатки для решения этой задачи, приведите примеры.
12. Какие задачи стремиться решить стратифицированный дизайн архитектуры приложения. Приведите пример.
13. Объясните назначение и способ применения шаблона несложной реализации при использовании стратифицированного дизайна. Приведите пример.

14. Объясните назначение и способ применения шаблона барьера из абстракций при использовании стратифицированного дизайна. Приведите пример.
15. Объясните назначение и способ применения шаблона минимального интерфейса при использовании стратифицированного дизайна. Приведите пример.
16. Объясните назначение и способ применения шаблона удобные слои при использовании стратифицированного дизайна. Приведите пример.
17. Объясните назначение и способ применения рефакторингов выделение явного аргумента и замены тела с помощью обратного вызова. Приведите примеры.
18. Какие возможности предоставляет использование функций как объектов первого рода. Для каждой возможности приведите аналогичный шаблон проектирования объектно-ориентированных приложений.
19. Приведите условия применения и пример использования рефакторингов выделения явного аргумента и замены тела с помощью обратного вызова для создания функций высшего порядка.
20. Опишите функцию высшего порядка `forEach`: способ вызова, подход к реализации, ограничения по использованию. Приведите примеры её использования.
21. Опишите функцию высшего порядка для формирования шаблонов копирования данных при записи и глубокого копирования. Приведите примеры их использования для массивов и ассоциативных массивов (объектов).
22. Опишите функцию высшего порядка `map`: способ вызова, подход к реализации, ограничения по использованию. Приведите примеры её использования.
23. Опишите функцию высшего порядка `filter`: способ вызова, подход к реализации, ограничения по использованию. Приведите примеры её использования.
24. Опишите функцию высшего порядка `reduce`: способ вызова, подход к реализации, ограничения по использованию. Приведите примеры её использования.
25. Опишите подходы к реализации цепочек по обработке цепочек: с оборачиванием функций высшего порядка и с выделением функций-вычислений. Укажите плюсы и минусы каждого из подходов. Приведите примеры.

## Критерии оценки

«Отлично» – ответ на вопросы показывает всестороннее знание темы, изученной литературы, изложен логично, аргументировано и в полном объеме. Основные понятия, выводы и обобщения сформулированы убедительно и доказательно. Продемонстрированы полные и глубокие навыки практического применения программного инструмента для обработки текста.

«Хорошо» – ответ на вопросы основан на твердом знании темы. Возможны недостатки в систематизации или в обобщении материала, неточности в выводах. Продемонстрированы хорошие навыки практического применения программного инструмента для обработки текста.

«Удовлетворительно» – ответ на вопросы базируется на знании основ предмета, но имеются значительные пробелы в изложении материала, затруднения в его изложении и систематизации, выводы слабо аргументированы, в содержании допущены теоретические ошибки. Продемонстрированы элементарные навыки практического применения программного инструмента для обработки текста для решения простых задач.

«Неудовлетворительно» – оценивается ответ на вопросы, в котором обнаружено неверное изложение темы, систематизации знаний, обобщений и выводов нет. Навыки

практического применения программного инструмента для обработки текста слабые и отрывочные или отсутствуют.