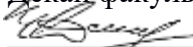


**МИНОБРНАУКИ РОССИИ**  
**Ярославский государственный университет им. П.Г. Демидова**

Кафедра вычислительных и программных систем

УТВЕРЖДАЮ  
Декан факультета ИВТ  
 Д.Ю. Чалый  
« 23 » мая 2023 г.

**Рабочая программа дисциплины**  
«Технологии функционального программирования в современных информационных системах»

**Направление подготовки**  
01.04.02 Прикладная математика и информатика

**Направленность (профиль)**  
«Математические основы искусственного интеллекта»

**Квалификация выпускника**  
Магистр

**Форма обучения**  
очная

Программа рассмотрена  
на заседании кафедры  
от «21» апреля 2023 г.,  
протокол № 8

Программа одобрена НМК  
факультета ИВТ  
протокол № 6 от  
«28» апреля 2023 г.

Ярославль

## 1. Цели освоения дисциплины

Дисциплина «ТЕХНОЛОГИИ ФУНКЦИОНАЛЬНОГО ПРОГРАММИРОВАНИЯ В СОВРЕМЕННЫХ ИНФОРМАЦИОННЫХ СИСТЕМАХ» относится к вариативной части ОП магистратуры

## 2. Место дисциплины в структуре ОП магистратуры

Дисциплина «Технологии функционального программирования в современных информационных системах» относится к вариативной части (дисциплина по выбору) ОП бакалавриата. В курсе рассматриваются вопросы получения специалистами компетенций в области автоматизации обработки естественных языков.

Содержание курса тесно связано фактически со всеми дисциплинами, которые изучались студентами. Освоению данной программы предшествуют учебные курсы по программированию и современным информационным технологиям.

Дисциплина «Технологии функционального программирования в современных информационных системах» знакомит магистрантов с важнейшими областями междисциплинарных исследований на стыке информатики и лингвистики. Дисциплина способствует профессиональному росту студентов, повышению их общеметодологического уровня, а также дальнейшему развитию навыков научно-исследовательской деятельности.

## 3. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения ОП магистратуры

Процесс изучения дисциплины направлен на формирование следующих элементов компетенций в соответствии с ФГОС ВО, ОП ВО и приобретения следующих знаний, умений, навыков и (или) опыта деятельности:

Код компетенции	Индикатор достижения компетенции (код и формулировка)	Перечень планируемых результатов обучения
<b>Профессиональные компетенции</b>		
ПК-1 Способен понимать и применять в научно-исследовательской и прикладной деятельности современный математический аппарат, основные законы естествознания, современные языки программирования и программное обеспечение; операционные системы и сетевые технологии.	ПК – 1.2 Применяет современные языки программирования, программное обеспечение для построения адекватных математических моделей решения задач в своей профессиональной деятельности.	Знать: <ul style="list-style-type: none"><li>• общие принципы построения систем автоматической обработки текста;</li><li>• основные формальные модели для уровней графематического, морфологического, синтаксического и семантического анализа текстовых массивов;</li><li>• основные алгоритмы автоматической обработки текстовых массивов;</li><li>• практические возможности современных систем автоматической обработки естественного языка.</li></ul> Уметь: <ul style="list-style-type: none"><li>• работать с системами морфологического и синтаксического анализа текста,</li></ul>

		<ul style="list-style-type: none"> <li>• осуществлять реферирование текста с использованием компьютерных средств,</li> <li>• моделировать морфологические и синтаксические структуры текста.</li> </ul> <p>Владеть:</p> <ul style="list-style-type: none"> <li>• методами математической лингвистики для анализа текстовой информации,</li> <li>• навыками работы с автоматическими словарями</li> <li>• принципами построения систем автоматического индексирования и реферирования тестовых массивов</li> </ul>
--	--	---

#### 4. Объем, структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 2 зач. ед., 72 акад. час.

№ п/п	Темы (разделы) дисциплины, их содержание	Семестр	Виды учебных занятий, включая самостоятельную работу студентов, и их трудоемкость (в академических часах)						Формы текущего контроля успеваемости  Форма промежуточной аттестации (по семестрам)
			Контактная работа						
			лекции	практические	лабораторные	консультации	аттестационные испытания	самостоятельная работа	
1.	Введение. Функциональные языки программирования (ФЯП). Области применения ФЯП, взаимодействие с другими парадигмами программирования.	1	2		2			7	

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.	1	2		2			7	
3.	Методы рефакторинга исходного кода ПО для уменьшения количества действий. Выделение вычислений из действий, отказ от неявных входов и выходов в действиях.	1	2		2			7	
4.	Методы обеспечения неизменяемости данных ПО: копирование при записи и глубокое копирование. Организация взаимодействия между изменяемыми и неизменяемыми данными внутри ПО.	1	2		2			7	
5.	Подходы к проектированию архитектуры приложений согласно стратифицированному подходу. Шаблоны несложной реализации, барьера из абстракций, минимального интерфейса и удобных слоёв.	1	2		2			7	
6.	Функции как объекты первого рода. Функциональное	1	6		6			2,7	

итерирование. Использование итераторов filter, map, reduce. Формирование цепочек по обработке данных.					2		
<b>Всего за 1 семестр</b>		16		16	2	37,7	Зачет
<b>Всего</b>		16		16	2	37,7	

#### Содержание разделов дисциплины:

Введение. Функциональные языки программирования (ФЯП). Области применения ФЯП, взаимодействие с другими парадигмами программирования.
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.
Методы рефакторинга исходного кода ПО для уменьшения количества действий. Выделение вычислений из действий, отказ от неявных входов и выходов в действиях.
Методы обеспечения неизменяемости данных ПО: копирование при записи и глубокое копирование. Организация взаимодействия между изменяемыми и неизменяемыми данными внутри ПО.
Подходы к проектированию архитектуры приложений согласно стратифицированному подходу. Шаблоны несложной реализации, барьера из абстракций, минимального интерфейса и удобных слоёв.
Функции как объекты первого рода. Функциональное итерирование. Использование итераторов filter, map, reduce. Формирование цепочек по обработке данных.

#### 5. Образовательные технологии, используемые при осуществлении образовательного процесса по дисциплине

В процессе обучения используются следующие образовательные технологии:

**Лекция-беседа** или «диалог с аудиторией», является наиболее распространенной и сравнительно простой формой активного вовлечения студентов в учебный процесс. Эта лекция предполагает непосредственный контакт преподавателя с аудиторией. Преимущество лекции-беседы состоит в том, что она позволяет привлекать внимание студентов к наиболее важным вопросам темы, определять содержание и темп изложения учебного материала с учетом особенностей студентов.

**Мастер-класс** – это особая форма учебного занятия, когда преподаватель-мастер передает свой опыт путем прямого и комментированного показа последовательности действий, методов, приемов и форм педагогической деятельности. Целью проведения мастер-класса является профессиональное, интеллектуальное и эстетическое воспитание студентов, и прежде всего, развитие в ходе мастер-класса способности студента самостоятельно и нестандартно мыслить.

**Лабораторная работа** – организация учебной работы с реальными материальными и информационными объектами, экспериментальная работа с аналоговыми моделями реальных объектов.

Практическое занятие – занятие, посвященное освоению конкретных умений и навыков и закреплению полученных на лекции знаний.

**6. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень лицензионного программного обеспечения и информационных справочных систем (при необходимости)**

В процессе осуществления образовательного процесса используются: для разработки документов, презентаций, для работы с электронными таблицами

OfficeStd 2013 RUS OLP NL Acdmc 021-10232

LibreOffice (свободное)

издательская система LaTeX;

– Среда разработки NetBeans 8.2 : [www.netbeans.org](http://www.netbeans.org). Доступ свободный

– OS Linux (свободная)

– для поиска учебной литературы библиотеки ЯрГУ – Автоматизированная библиотечная информационная система "БУКИ-NEXT" (АБИС "Буки-Next").

**7. Перечень основной и дополнительной учебной литературы, ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины**

а) основная:

Пентус, А. Е. Математическая теория формальных языков / Пентус А. Е. , Пентус М. Р. - Москва : Национальный Открытый Университет "ИНТУИТ", 2016. (Основы информатики и математики) - ISBN 5-9556-0062-0. - Текст : электронный // ЭБС "Консультант студента" : [сайт]. - URL : <https://www.studentlibrary.ru/book/ISBN5955600620.html>

в) ресурсы сети «Интернет»

- Автоматическая обработка текстов на естественном языке. Учебное пособие. / Большакова, Е.И.; Клышинский, Э.С.; Ландэ, Д.В.; Носков, А.А.; Пескова, О.В.; Ягунова, Е.В. [Электронный ресурс] <https://nsu.ru/xmlui/handle/nsu/8982>

Электронно-библиотечная система «Юрайт»( <https://urait.ru/> ).

Электронно-библиотечная система «Лань»( <https://e.lanbook.com/>).

**8. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине**

Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине включает в свой состав специальные помещения:

-учебные аудитории для проведения занятий лекционного типа и практических занятий (семинаров);

- учебные аудитории для проведения групповых и индивидуальных консультаций,

- учебные аудитории для проведения текущего контроля и промежуточной аттестации;

-помещения для самостоятельной работы;

-помещения для хранения и профилактического обслуживания технических средств обучения.

Специальные помещения укомплектованы средствами обучения, служащими для представления учебной информации большой аудитории.

Для проведения занятий лекционного типа предлагаются наборы демонстрационного оборудования и учебно-наглядных пособий, хранящиеся на электронных носителях и обеспечивающие тематические иллюстрации, соответствующие рабочим программам дисциплин.

Помещения для лабораторных занятий и самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду организации.

Число посадочных мест в лекционной аудитории больше либо равно списочному составу потока, а в аудитории для практических занятий (семинаров) – списочному составу группы обучающихся.

**Приложение №1 к рабочей программе дисциплины  
«Технологии функционального программирования  
в современных информационных системах»**

**Фонд оценочных средств**

**для проведения текущей и промежуточной аттестации студентов  
по дисциплине**

**1. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций**

**1.1. Контрольные задания и иные материалы, используемые в процессе текущей аттестации**

**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 библиотечками. 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 программирования Python	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://docs.python.org/3/">https://docs.python.org/3/</a></li> <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	<a href="https://www.haskell.org/documentation/">https://www.haskell.org/documentation/</a>



		способов взаимодействия с библиотеками на языке программирования Си.	
6	Анализ средств применения функциональных инструментов в языке программирования Erlang	Выяснение списка доступных функциональных инструментов, предоставляемых средствами разработки и доступными библиотеками. 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 библиотеками. 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 библиотеками. 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 стремятся 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 выделение 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.

## **2. Перечень компетенций, этапы их формирования, описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкалы оценивания**

### **2.1. Шкала оценивания сформированности компетенций и ее описание**

Оценивание уровня сформированности компетенций в процессе освоения дисциплины осуществляется по следующей трехуровневой шкале:

*Пороговый уровень* - предполагает отражение тех ожидаемых результатов, которые определяют минимальный набор знаний и (или) умений и (или) навыков, полученных студентом в результате освоения дисциплины. Пороговый уровень является обязательным уровнем для студента к моменту завершения им освоения данной дисциплины.

*Продвинутый уровень* - предполагает способность студента использовать знания, умения, навыки и (или) опыт деятельности, полученные при освоении дисциплины, для решения профессиональных задач. Продвинутый уровень превосходит пороговый уровень по нескольким существенным признакам.

*Высокий уровень* - предполагает способность студента использовать потенциал интегрированных знаний, умений, навыков и (или) опыта деятельности, полученных при освоении дисциплины, для творческого решения профессиональных задач и самостоятельного поиска новых подходов в их решении путем комбинирования и использования известных способов решения применительно к конкретным условиям. Высокий уровень превосходит пороговый уровень по всем существенным признакам.

## 2.2. Перечень компетенций, этапы их формирования, описание показателей и критериев оценивания компетенций на различных этапах их формирования

Код компетенции	Форма контроля	Этапы формирования (№ темы (раздела))	Показатели оценивания	Шкала и критерии оценивания компетенций на различных этапах их формирования		
				Пороговый уровень	Продвинутый уровень	Высокий уровень
<b>Профессиональные компетенции</b>						
ПК-3	Индивидуальные задания Лабораторная работа. Зачет.	1-8	<p>Знать:</p> <ul style="list-style-type: none"> <li>• общие принципы построения систем автоматической обработки текста;</li> <li>• основные формальные модели для уровней графематического, морфологического, синтаксического и семантического анализа текстовых массивов;</li> <li>• основные алгоритмы автоматической обработки текстовых массивов;</li> <li>• практические возможности современных систем</li> </ul>	<p>обучающийся должен:</p> <p><i>а) знать:</i></p> <ul style="list-style-type: none"> <li>• основные цели и задачи компьютерной лингвистики;</li> <li>• основные методы и подходы к автоматической обработке текстов (правилловые, статистические);</li> <li>• основные классы приложений, развиваемых на базе компьютерной лингвистики (информационный поиск, автоматический и автоматизированный перевод текстов);</li> </ul>	<p>обучающийся должен:</p> <p><i>а) знать:</i></p> <ul style="list-style-type: none"> <li>• основные цели и задачи компьютерной лингвистики;</li> <li>• основные методы и подходы к автоматической обработке текстов (правилловые, статистические, в т.ч. машинное обучение, гибридные);</li> <li>• основные классы приложений, развиваемых на базе компьютерной лингвистики (информационный поиск, глубокий анализ данных, автоматический и автоматизированный перевод текстов с одного языка на другой, автоматическое аннотирование и</li> </ul>	<p>обучающийся должен:</p> <p><i>а) знать:</i></p> <ul style="list-style-type: none"> <li>• основные цели и задачи компьютерной лингвистики;</li> <li>• основные методы и подходы к автоматической обработке текстов (правилловые, статистические, в т.ч. машинное обучение, гибридные);</li> <li>• основные классы приложений, развиваемых на базе компьютерной лингвистики (информационный поиск, глубокий анализ данных, автоматический и автоматизированный перевод текстов с одного языка на другой, автоматическое аннотирование и</li> </ul>

			<p>автоматической обработки естественного языка.</p> <p>Уметь:</p> <ul style="list-style-type: none"> <li>• работать с системами морфологического и синтаксического анализа текста,</li> <li>• осуществлять реферирование текста с использованием компьютерных средств,</li> <li>• моделировать морфологические и синтаксические структуры текста.</li> </ul> <p>Владеть:</p> <ul style="list-style-type: none"> <li>• методами математической лингвистики для анализа текстовой информации,</li> <li>• навыками работы с автоматическими словарями</li> <li>• принципами построения систем автоматического индексирования и</li> </ul>	<ul style="list-style-type: none"> <li>• основные классы цифровых лингвистических ресурсов, создаваемых методами компьютерной лингвистики (компьютерные словари);</li> </ul> <p><i>б) уметь:</i></p> <ul style="list-style-type: none"> <li>• строить базовые правила систем автоматической обработки текстов;</li> <li>• строить базовые морфологические и синтаксические структуры предложения (на примере русского и английского языков);</li> </ul> <p><i>в) владеть:</i></p> <ul style="list-style-type: none"> <li>• принципами использования электронных языковых ресурсов (текстовых, речевых и мультимодальных корпусов; словарей, тезаурусов и онтологий).</li> </ul>	<p>одного языка на другой, автоматическое аннотирование и реферирование документов);</p> <ul style="list-style-type: none"> <li>• основные классы цифровых лингвистических ресурсов, создаваемых методами компьютерной лингвистики (компьютерные одноязычные и многоязычные словари);</li> </ul> <p><i>б) уметь:</i></p> <ul style="list-style-type: none"> <li>• строить базовые правила систем автоматической обработки текстов;</li> <li>• разбираться в правилах и алгоритмах автоматической обработки текстов;</li> <li>• строить базовые морфологические и синтаксические структуры предложения (на примере русского и английского языков);</li> </ul>	<p>реферирование документов, анализ тональности текста, человеко-машинное общение на естественном языке);</p> <ul style="list-style-type: none"> <li>• основные классы цифровых лингвистических ресурсов, создаваемых методами компьютерной лингвистики (компьютерные одноязычные и многоязычные словари, аннотированные корпуса текстов);</li> </ul> <p><i>б) уметь:</i></p> <ul style="list-style-type: none"> <li>• строить базовые правила систем автоматической обработки текстов;</li> <li>• разбираться в правилах и алгоритмах автоматической обработки текстов;</li> <li>• строить базовые морфологические и синтаксические структуры предложения (на примере русского и английского языков);</li> <li>• использовать лингвистические</li> </ul>
--	--	--	---	---	--	---

			реферирования тестовых массивов.		<p><i>в) владеть:</i></p> <ul style="list-style-type: none"> <li>• навыком освоения большого объема информации;</li> <li>• принципами создания электронных языковых ресурсов (текстовых, речевых и мультимодальных корпусов; словарей, тезаурусов и онтологий) и умение пользоваться такими ресурсами.</li> </ul>	<p>технологии для проектирования систем автоматической обработки письменного текста на естественном языке</p> <p><i>в) владеть:</i></p> <ul style="list-style-type: none"> <li>• навыком освоения большого объема информации;</li> <li>• принципами создания электронных языковых ресурсов (текстовых, речевых и мультимодальных корпусов; словарей, тезаурусов и онтологий; фонетических, лексических, грамматических и иных баз данных и баз знаний) и умение пользоваться такими ресурсами;</li> <li>• навыками постановки научно-исследовательских задач и навыками самостоятельной работы.</li> </ul>
--	--	--	----------------------------------	--	---	--

**Общекультурные компетенции**

ОК-1	Индивидуальные задания Лабораторная работа. Зачет.	1-8	<p>Знать:</p> <ul style="list-style-type: none"> <li>• методику проведения исследований.</li> </ul> <p>Уметь:</p>	<p>обучающийся должен:</p> <p><i>а) знать:</i></p> <ul style="list-style-type: none"> <li>• основные методы и подходы к автоматической обработке текстов</li> </ul>	<p>обучающийся должен:</p> <p><i>а) знать:</i></p> <ul style="list-style-type: none"> <li>• основные методы и подходы к автоматической обработке текстов</li> </ul>	<p>обучающийся должен:</p> <p><i>а) знать:</i></p> <ul style="list-style-type: none"> <li>• различные методы и подходы к автоматической обработке текстов (правилловые, статистические, в т.ч.</li> </ul>
------	--	-----	---	---	---	---



			<ul style="list-style-type: none"> <li>• выполнять исследовательские эксперименты</li> </ul> <p>Владеть:</p> <ul style="list-style-type: none"> <li>• методами математической лингвистики для анализа текстовой информации</li> </ul>	<p>(правилловые, статистические);</p> <p><i>б) уметь:</i></p> <ul style="list-style-type: none"> <li>• использовать лингвистические технологии для проектирования систем автоматической обработки письменного текста на естественном языке</li> </ul> <p><i>в) владеть:</i></p> <ul style="list-style-type: none"> <li>• навыками самостоятельной работы.</li> </ul>	<p>(правилловые, статистические);</p> <p><i>б) уметь:</i></p> <ul style="list-style-type: none"> <li>• использовать лингвистические технологии для проектирования систем автоматической обработки письменного текста на естественном языке</li> <li>• проводить эксперименты по обработке текста</li> </ul> <p><i>в) владеть:</i></p> <ul style="list-style-type: none"> <li>• навыками постановки научно-исследовательских задач и навыками самостоятельной работы.</li> </ul>	<p>машинное обучение, гибридные);</p> <p><i>б) уметь:</i></p> <ul style="list-style-type: none"> <li>• использовать лингвистические технологии для проектирования систем автоматической обработки письменного текста на естественном языке</li> <li>• проводить эксперименты по обработке текста</li> <li>• рассчитывать статистические характеристики эксперимента</li> </ul> <p><i>в) владеть:</i></p> <ul style="list-style-type: none"> <li>• навыками постановки научно-исследовательских задач и навыками самостоятельной работы.</li> </ul>
--	--	--	---	--	---	--

### **3. Методические рекомендации преподавателю по процедуре оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций**

Целью процедуры оценивания является определение степени овладения студентом ожидаемыми результатами обучения (знаниями, умениями, навыками и (или) опытом деятельности).

Процедура оценивания степени овладения студентом ожидаемыми результатами обучения осуществляется с помощью методических материалов, представленных в разделе «Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций»

#### **3.1 Критерии оценивания степени овладения знаниями, умениями, навыками и (или) опытом деятельности, определяющие уровни сформированности компетенций**

Пороговый уровень (общие характеристики):

- владение основным объемом знаний по программе дисциплины;
- знание основной терминологии данной области знаний, стилистически грамотное, логически правильное изложение ответа на вопросы без существенных ошибок;
- владение инструментарием дисциплины, умение его использовать в решении стандартных (типовых) задач;
- способность самостоятельно применять типовые решения в рамках рабочей программы дисциплины;
- усвоение основной литературы, рекомендованной рабочей программой дисциплины;
- знание базовых теорий, концепций и направлений по изучаемой дисциплине;
- самостоятельная работа на практических и лабораторных занятиях, периодическое участие в групповых обсуждениях, достаточный уровень культуры исполнения заданий.

Продвинутый уровень (общие характеристики):

- достаточно полные и систематизированные знания в объёме программы дисциплины;
- использование основной терминологии данной области знаний, стилистически грамотное, логически правильное изложение ответа на вопросы, умение делать выводы;
- владение инструментарием дисциплины, умение его использовать в решении учебных и профессиональных задач;
- способность самостоятельно решать сложные задачи (проблемы) в рамках рабочей программы дисциплины;
- усвоение основной и дополнительной литературы, рекомендованной рабочей программой дисциплины;
- умение ориентироваться в базовых теориях, концепциях и направлениях по изучаемой дисциплине и давать им сравнительную оценку;
- самостоятельная работа на практических и лабораторных занятиях, участие в групповых обсуждениях, высокий уровень культуры исполнения заданий.

Высокий уровень (общие характеристики):

- систематизированные, глубокие и полные знания по всем разделам дисциплины;
- точное использование терминологии данной области знаний, стилистически грамотное, логически правильное изложение ответа на вопросы, умение делать обоснованные выводы;

- безупречное владение инструментарием дисциплины, умение его использовать в постановке и решении научных и профессиональных задач;
- способность самостоятельно и творчески решать сложные задачи (проблемы) в рамках рабочей программы дисциплины;
- полное и глубокое усвоение основной и дополнительной литературы, рекомендованной рабочей программой дисциплины;
- умение ориентироваться в основных теориях, концепциях и направлениях по изучаемой дисциплине и давать им критическую оценку;
- активная самостоятельная работа на практических и лабораторных занятиях, творческое участие в групповых обсуждениях, высокий уровень культуры исполнения заданий.

### **3.2 Описание процедуры выставления оценки**

В зависимости от уровня сформированности каждой компетенции по окончании освоения дисциплины студенту выставляется оценка «зачтено», «незачтено».

Показатели и критерии, используемые при выставлении оценки подробно описаны в разделе «Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций».

Высокий уровень формирования компетенций соответствует оценке «отлично» за индивидуальное задание и лабораторную работу.

Продвинутый уровень формирования компетенций соответствует оценке «хорошо» за индивидуальное задание и лабораторную работу.

Пороговый уровень формирования компетенций соответствует оценке «удовлетворительно» за индивидуальное задание и лабораторную работу.

Оценка «отлично» выставляется студенту, у которого каждая компетенция (полностью или частично формируемая данной дисциплиной) сформирована на высоком уровне.

Оценка «хорошо» выставляется студенту, у которого каждая компетенция (полностью или частично формируемая данной дисциплиной) сформирована не ниже, чем на продвинутом уровне.

Оценка «удовлетворительно» выставляется студенту, у которого каждая компетенция (полностью или частично формируемая данной дисциплиной) сформирована не ниже, чем на пороговом уровне.

Оценка «неудовлетворительно» выставляется студенту, у которого хотя бы одна компетенция (полностью или частично формируемая данной дисциплиной) сформирована ниже, чем на пороговом уровне.

Оценка «зачет» выставляется студенту, у которого каждая компетенция (полностью или частично формируемая данной дисциплиной) сформирована не ниже, чем на пороговом уровне.

Оценка «незачтено» выставляется студенту, у которого хотя бы одна компетенция (полностью или частично формируемая данной дисциплиной) сформирована ниже, чем на пороговом уровне.

## **Приложение №2 к рабочей программе дисциплины «Технологии функционального программирования в современных информационных системах»**

### **Методические указания для студентов по освоению дисциплины**

Занятия проводятся в интерактивной форме с использованием мультимедиа-технологий. Занятия предполагают наличие дискуссий по поводу тех или иных вопросов разработки программных приложений осуществляемых в результате соответствующего предложения преподавателя.

Практическое применение полученных знаний отрабатывается и во время лабораторных занятий, ориентированных помимо закрепления лекционного материала на разбор различных модельных ситуаций, характерных для разработки современных программных продуктов. Для успешного освоения дисциплины очень важно решение достаточно большого количества задач, как в аудитории, так и самостоятельно в качестве домашних заданий. В основном такими задачами являются лабораторные работы различного объема. Примеры решения задач разбираются на занятиях. Для решения всех задач необходимо знать и понимать лекционный материал. Поэтому в процессе изучения дисциплины рекомендуется регулярное повторение пройденного лекционного материала. Материал, законспектированный на лекциях, необходимо дома еще раз прорабатывать и при необходимости дополнять информацией, полученной на консультациях, практических занятиях или из учебной литературы.

Текущий контроль успеваемости осуществляется в форме опросов по основным понятиям и концепциям курса и защиты реферата по заданной теме, осуществляемый в ходе лабораторных занятий. Для самостоятельной работы студентам предлагается разрабатывать и реализовывать алгоритмы создания компьютерных систем с применением разобранных во время лекций и лабораторных занятий подходов и методик. Окончательная аттестация осуществляется в форме зачета, основную часть которого составляют результаты лабораторных работ, а также собеседование по тематике курса.

### **Учебно-методическое обеспечение самостоятельной работы студентов по дисциплине**

Для самостоятельной работы особенно рекомендуется использовать учебную литературу, указанную в разделе № 7 данной рабочей программы.

Также для подбора учебной литературы рекомендуется использовать широкий спектр интернет-ресурсов:

1. Электронно-библиотечная система «Университетская библиотека online» ([www.biblioclub.ru](http://www.biblioclub.ru)) - электронная библиотека, обеспечивающая доступ к наиболее востребованным материалам-первоисточникам, учебной, научной и художественной литературе ведущих издательств (\*регистрация в электронной библиотеке – только в сети университета. После регистрации работа с системой возможна с любой точки доступа в Internet.).

2. Для самостоятельного подбора литературы в библиотеке ЯрГУ рекомендуется использовать:

1. Личный кабинет ([http://lib.uniyar.ac.ru/opac/bk\\_login.php](http://lib.uniyar.ac.ru/opac/bk_login.php)) дает возможность получения on-line доступа к списку выданной в автоматизированном режиме литературы, просмотра и копирования электронных версий изданий сотрудников университета (учеб. и метод. пособия, тексты лекций и т.д.) Для работы в «Личном кабинете» необходимо зайти на сайт Научной библиотеки ЯрГУ с любой точки, имеющей доступ в Internet, в пункт меню «Электронный каталог»; пройти процедуру авторизации, выбрав вкладку «Авторизация», и заполнить представленные поля информации.

2. Электронная библиотека учебных материалов ЯрГУ ([http://www.lib.uniyar.ac.ru/opac/bk\\_cat\\_find.php](http://www.lib.uniyar.ac.ru/opac/bk_cat_find.php)) содержит более 2500 полных текстов учебных и учебно-методических материалов по основным изучаемым

дисциплинам, изданных в университете. Доступ в сети университета, либо по логину/паролю.

3. Электронная картотека [«Книгообеспеченность»](#) ([http://www.lib.uniyar.ac.ru/opac/bk\\_bookreq\\_find.php](http://www.lib.uniyar.ac.ru/opac/bk_bookreq_find.php)) раскрывает учебный фонд научной библиотеки ЯрГУ, предоставляет оперативную информацию о состоянии книгообеспеченности дисциплин основной и дополнительной литературой, а также цикла дисциплин и специальностей. Электронная картотека [«Книгообеспеченность»](#) доступна в сети университета и через Личный кабинет.