

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования «Новосибирский национальный исследовательский
государственный университет» (Новосибирский государственный университет, НГУ)

Факультет информационных технологий

СОГЛАСОВАНО

Декан ФИТ НГУ

 М.М. Лаврентьев

«25» апреля 2023 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Современные методы программирования

Направление подготовки: 09.04.01 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА

Направленность (профиль): Искусственный интеллект и Data Science

Форма обучения: очная

Год обучения: 1, семестр: 1

№	Вид деятельности	Семестр
		1
1	Лекции, час.	64
2	Практические занятия, час.	
3	Лабораторные занятия, час.	32
4	Занятий в контактной форме без учета промежуточной аттестации, час, из них	98
5	в электронной форме, час.	
6	из них аудиторных занятий, час.	96
7	из них в активной и интерактивной форме, час.	96
8	консультаций, час.	2
9	Самостоятельная работа, час.	152
10	в том числе на выполнение письменных работ, час	90
11	Форма аттестации (экзамен, зачет, дифференцированный зачет), час	Э 2
12	Всего зачетных единиц ¹	7

Новосибирск 2023

¹ С учетом выделенных часов на промежуточную аттестацию

Рабочая программа дисциплины составлена на основании федерального государственного образовательного стандарта (ФГОС) высшего образования - магистратура по направлению подготовки 09.04.01 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА.

Федеральный государственный образовательный стандарт (ФГОС) высшего образования - магистратура по направлению подготовки 09.04.01 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА введен в действие приказом Минобрнауки от 19.09.2017 № 918.

Место дисциплины в структуре учебного плана: Блок 1 Дисциплины (модули); обязательная часть, обязательная дисциплина

Рабочая программа дисциплины утверждена решением Ученого совета факультета информационных технологий от 24.04.2023, протокол №91.

Программу разработал:

доцент кафедры систем информатики ФИТ,
кандидат физико-математических наук



Д.С. Мигинский

заведующий кафедрой систем информатики ФИТ,
доктор физико-математических наук



М.М. Лаврентьев

Ответственный за образовательную программу:

заведующий кафедрой систем информатики ФИТ,
доктор физико-математических наук



М.М. Лаврентьев

Аннотация к рабочей программе дисциплины «Современные методы программирования»

Дисциплина «Современные методы программирования» реализуется в рамках образовательной программы высшего образования – программы магистратуры 09.04.01 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА. ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ И DATA SCIENCE по очной форме обучения на русском языке.

Место в образовательной программе:

Дисциплина «Современные методы программирования» является базовой для работы в рамках практики и выполнения выпускной квалификационной работы.

Содержание дисциплины охватывает круг вопросов, связанных с современными языками программирования, их классификацией и областями их применения; различными методами абстрагирования, обеспечения модульности и других аспектов проектирования программных систем

Дисциплина «Современные методы программирования» реализуется в первом семестре в рамках обязательной части дисциплин (модулей) Блока 1, является обязательной дисциплиной.

Дисциплина «Современные методы программирования» направлена на формирование компетенций:

Способен разрабатывать оригинальные алгоритмы и программные средства, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач (ОПК-2), в части следующих индикаторов достижения компетенции:

ОПК-2.1. Знать: современные информационно-коммуникационные и интеллектуальные технологии, инструментальные среды, программно-технические платформы для решения профессиональных задач

ОПК-2.2 Уметь: обосновывать выбор современных информационно-коммуникационных и интеллектуальных технологий, разрабатывать оригинальные программные средства для решения профессиональных задач

ОПК-2.3. Владеть: методами разработки оригинальных программных средств, в том числе с использованием современных информационно-коммуникационных и интеллектуальных технологий, для решения профессиональных задач

Способен разрабатывать и модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем (ОПК-5) , в части следующих индикаторов достижения компетенции:

ОПК-5.2. Уметь: модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем для решения профессиональных задач

ОПК-5.3. Владеть: навыками разработки программного и аппаратного обеспечения информационных и автоматизированных систем для решения профессиональных задач

Способен разрабатывать компоненты программно-аппаратных комплексов обработки информации и автоматизированного проектирования (ОПК-6) , в части следующих индикаторов достижения компетенции:

ОПК-6.1. Знать: аппаратные средства и платформы инфраструктуры информационных технологий, виды, назначение, архитектуру, методы разработки и администрирования программно-аппаратных комплексов объекта профессиональной деятельности

ОПК-6.2. Уметь: анализировать техническое задание, разрабатывать и оптимизировать программный код для решения задач обработки информации и автоматизированного проектирования

ОПК-6.3 Владеть: навыками составления технической документации по использованию и настройке компонентов программно-аппаратного комплекса

Способен адаптировать зарубежные комплексы обработки информации и автоматизированного проектирования к нуждам отечественных предприятий (ОПК-7), в части следующих индикаторов достижения компетенции:

ОПК-7.3. Владеть: навыками настройки интерфейса, разработки пользовательских шаблонов, подключения библиотек, добавления новых функций

Способен осуществлять эффективное управление разработкой программных средств и проектов (ОПК-8) , в части следующих индикаторов достижения компетенции:

ОПК-8.1. Знать: методы и средства разработки программного обеспечения, методы управления проектами разработки программного обеспечения, способы организации проектных данных, нормативно-технические документы (стандарты и регламенты) по разработке программных средств и проектов

Перечень основных разделов дисциплины:

Преподавание дисциплины предусматривает проведение следующих видов учебных занятий: лекции, лабораторные работы, самостоятельную работу студента.

Основные темы:

Раздел 1: Современные динамические языки

Раздел 2. Современные функциональные языки

Раздел 3. Элементы метапрограммирования, аспектно-ориентированное программирование.

Предусмотрено проведение лекций, лабораторных и практических занятий в дистанционном формате.

Общий объем дисциплины – 7 зачетные единицы (252 часа)

Правила аттестации по дисциплине.

Текущая аттестация по дисциплине «Современные методы программирования» проводится в форме портфолио. Промежуточная аттестация проводится в формате экзамена.

В рамках самостоятельной работы обучающийся выполняет задания, а также занимается изучением необходимых для этого языков программирования (на лекциях даются только основы).

Задания для лабораторных занятий и самостоятельной работы делятся на два класса:

1. обязательные задания небольшой сложности (№№1-7);
2. более сложные дополнительные задания (№8).

Обязательные задания должны быть сданы в течение 2 недель с момента, как они были сформулированы на лекции, в противном случае оценка за задачу снижается. Выполнение обязательных заданий является критерием допуска к экзамену. Если обучающийся претендует на оценку выше чем «удовлетворительно», он должен выполнить (полностью или частично) дополнительное задание, формулировка которого уточняется с преподавателем индивидуально.

Рекомендуется выполнение дополнительных заданий попарно, с четким разделением границ ответственности каждого студента. В разделе 7.3.1 приведены примерные постановки дополнительных заданий. Студент может предложить свою постановку. По результатам выполнения заданий ставится предварительная оценка.

В конце семестра проводится устный экзамен, на котором студенту предлагается билет с двумя вопросами по материалам лекций.

Выполнение всех обязательных заданий является необходимым и достаточным условием для получения допуска к экзамену с предварительной оценкой «удовлетворительно».

Промежуточная аттестация:

Промежуточная аттестация по курсу проводится по результатам сдачи устного экзамена. Условием допуска к экзамену является получение положительной предварительной оценки (по результатам выполнения заданий).

Оценка за экзамен в свою очередь складывается из предварительной оценки и ответов на экзаменационные вопросы. При этом предварительная оценка может быть улучшена или ухудшена не более чем на 1 балл. Таким образом, студент имеет право отказаться отвечать на экзаменационные вопросы и получить оценки на балл ниже предварительной.

Промежуточная аттестация по дисциплине производится: в 1 семестре в виде экзамена. По результатам аттестации выставляется оценка по шкале «неудовлетворительно», «удовлетворительно», «хорошо», «отлично». Оценки «отлично», «хорошо», «удовлетворительно» означают успешное прохождение промежуточной аттестации.

Учебно-методическое обеспечение дисциплины.

Учебно-методические материалы по дисциплине «Современные методы программирования» выложены на странице курса в сети Интернет, создаваемой для каждого нового набора

1. Внешние требования к дисциплине

Таблица 1.1

<p>Компетенция ОПК-2. Способен разрабатывать оригинальные алгоритмы и программные средства, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач, в части следующих индикаторов достижения компетенции:</p>	
ОПК-2.1.	Знать: современные информационно-коммуникационные и интеллектуальные технологии, инструментальные среды, программно-технические платформы для решения профессиональных задач
ОПК-2.2	Уметь: обосновывать выбор современных информационно-коммуникационных и интеллектуальных технологий, разрабатывать оригинальные программные средства для решения профессиональных задач
ОПК-2.3.	Владеть: методами разработки оригинальных программных средств, в том числе с использованием современных информационно-коммуникационных и интеллектуальных технологий, для решения профессиональных задач
<p>Компетенция ОПК-5. Способен разрабатывать и модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем, в части следующих индикаторов достижения компетенции:</p>	
ОПК-5.2.	Уметь: модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем для решения профессиональных задач
ОПК-5.3.	Владеть: навыками разработки программного и аппаратного обеспечения информационных и автоматизированных систем для решения профессиональных задач
<p>Компетенция ОПК-6. Способен разрабатывать компоненты программно-аппаратных комплексов обработки информации и автоматизированного проектирования, в части следующих индикаторов достижения компетенции:</p>	
ОПК-6.1.	Знать: аппаратные средства и платформы инфраструктуры информационных технологий, виды, назначение, архитектуру, методы разработки и администрирования программно-аппаратных комплексов объекта профессиональной деятельности
ОПК-6.2.	Уметь: анализировать техническое задание, разрабатывать и оптимизировать программный код для решения задач обработки информации и автоматизированного проектирования
ОПК-6.3	Владеть: навыками составления технической документации по использованию и настройке компонентов программно-аппаратного комплекса
<p>Компетенция ОПК-7. Способен адаптировать зарубежные комплексы обработки информации и автоматизированного проектирования к нуждам отечественных предприятий, в части следующих индикаторов достижения компетенции:</p>	
ОПК-7.3.	Владеть: навыками настройки интерфейса, разработки пользовательских шаблонов, подключения библиотек, добавления новых функций
<p>Компетенция ОПК-8. Способен осуществлять эффективное управление разработкой программных средств и проектов, в части следующих индикаторов достижения компетенции:</p>	
ОПК-8.1.	Знать: методы и средства разработки программного обеспечения, методы управления проектами разработки программного обеспечения, способы организации проектных данных, нормативно-технические документы

	(стандарты и регламенты) по разработке программных средств и проектов
--	---

2. Требования к результатам освоения дисциплины

Таблица 2.1

Результаты изучения дисциплины по уровням освоения (иметь представление, знать, уметь, владеть)	Формы организации занятий		
	Лекции	Практики / семинары	Самостояте льная работа
ОПК-2.1. Знать: современные информационно-коммуникационные и интеллектуальные технологии, инструментальные среды, программно-технические платформы для решения профессиональных задач			
1. Знать методы и языки объектно-ориентированного, функционального, аспектно-ориентированного программирования	+	+	+
ОПК-2.2 Уметь: обосновывать выбор современных информационно-коммуникационных и интеллектуальных технологий, разрабатывать оригинальные программные средства для решения профессиональных задач			
2. Уметь комбинировать различные языки и системы программирования, а также методы проектирования с целью оптимального решения поставленных задач проектирования ПО	+	+	+
ОПК-2.3. Владеть: методами разработки оригинальных программных средств, в том числе с использованием современных информационно-коммуникационных и интеллектуальных технологий, для решения профессиональных задач			
3. Владеть методами объектно-ориентированного, функционального, аспектно-ориентированного программирования	+	+	+
ОПК-5.2. Уметь: модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем для решения профессиональных задач			
4. Уметь разворачивать среды и инструменты разработки и сборки для различных языков и систем программирования		+	+
ОПК-5.3. Владеть: навыками разработки программного и аппаратного обеспечения информационных и автоматизированных систем для решения профессиональных задач			
5. Владеть навыками разработки системного программного обеспечения: фреймворков и систем программирования	+	+	+
ОПК-6.1. Знать: аппаратные средства и платформы инфраструктуры информационных технологий, виды, назначение, архитектуру, методы разработки и администрирования программно-аппаратных комплексов объекта профессиональной деятельности			
6. Знать методы разработки, настройки и комбинирования систем программирования	+		
ОПК-6.2. Уметь: анализировать техническое задание, разрабатывать и оптимизировать программный код для решения задач обработки информации и автоматизированного проектирования			
7. Уметь разрабатывать архитектуру и программный код для системного программного обеспечения	+	+	+
ОПК-6.3 Владеть: навыками составления технической документации по использованию и настройке компонентов программно-аппаратного комплекса			
8. Владеть навыками разработки технической документации на системное программное обеспечение		+	+
ОПК-7.3. Владеть: навыками настройки интерфейса, разработки пользовательских шаблонов,			

подключения библиотек, добавления новых функций			
9. Владеть навыками настройки современных сред разработки под заданные системы программирования и библиотеки		+	+
ОПК-8.1. Знать: методы и средства разработки программного обеспечения, методы управления проектами разработки программного обеспечения, способы организации проектных данных, нормативно-технические документы (стандарты и регламенты) по разработке программных средств и проектов			
10. Знать современные методы проектирования, разработки, документирования и организации проектной деятельности в области программного обеспечения	+	+	+

3. Содержание и структура учебной дисциплины

Таблица 3.1

Темы лекций	Активные формы, час.	Часы	Ссылки на результаты обучения
Семестр: 1			
<p>Раздел 1: Современные динамические языки</p> <p>1.1. Общая классификация языков по назначению и модели исполнения. Общие свойства динамических языков (динамическая типизация, модель трансляции и исполнения)</p> <p>1.2. Язык Ruby: основные конструкции языка, коллекции. Функциональный стиль программирования в Ruby: блоки и замыкания. Итераторы. Реорганизующее присваивание.</p> <p>1.3. Особенности объектной модели Ruby: унифицированность объектного представления, модули и примеси, инкапсуляция. Динамическое изменение классов, элементы Meta-Object Protocol (MOP) в Ruby. JRuby и взаимодействие с Java, Java Scripting API.</p> <p>1.4. Регулярные выражения. Классы символов. Жадные и нежадные выражения. Основные операции с регулярными выражениями.</p>	12	12	1,3,7,10
<p>Раздел 2. Современные функциональные языки</p> <p>2.1. Классификация языков по парадигмам программирования. Функциональное программирование (ФП). Неподвижное состояние объекта как ключевое отличие ФП от ООП. Чистые функции, функции высших порядков. Функции, как объекты первого класса. Лексические контексты, анонимные функции, замыкания. Основные семейства функциональных языков. Историческая связь динамических и функциональных языков.</p> <p>2.2. Общие характеристики семейства языков Lisp: единое представление кода и данных, S-выражения, модель трансляции и исполнения, REPL. Язык Clojure, как современный представитель семейства Lisp: основные структуры языка. Компонентное тестирование в Clojure.</p>	20	20	1,3,10

2.3. Функциональные возможности Clojure: коллекции, реорганизуемое присваивание, мемоизация, отложенные вычисления, бесконечные структуры данных. Абстрагирование данных с помощью функциональных примитивов. Моделирование времени с помощью потоков. Символьные вычисления. Преимущества и недостатки ФП в сравнении с ООП.			
2.4. Императивные возможности Clojure. Software Transactional Memory. Многопоточность. Ссылки, атомы, агенты, переменные, виды транзакций. Взаимодействие с Java.			
Раздел 3. Элементы метапрограммирования, аспектно-ориентированное программирование.			1,2,3,5,6,7,10
3.1. Управляемая кодогенерация. Макросы в Lisp (на примере Clojure). Модель исполнения макросов. Макросы, как способ расширения языка.			
3.2. Понятие о проблемно-специфичных языках (DSL) и языках сценариев. Методы построения и генерации DSL.			
3.3. Динамические объектные модели. CLOS: обобщенный динамический полиморфизм, обобщенные функции и мультиметоды, вспомогательные методы. Реализация элементов CLOS в Clojure. Интроспекция, введение в MOP.	32	32	
3.4. Сквозная функциональность (cross-cutting concerns), проблема модульности. Традиционные методы обеспечения модульности в условиях сквозной функциональности.			
3.5. Аспектно-ориентированное программирование (АОП). Динамические лексические контексты, их реализация в Clojure. Элементы АОП в CLOS.			
3.6. Применение АОП в проектирование. Преимущества и недостатки по сравнению с традиционными методами проектирования. Примеры задач, эффективно решаемых с помощью АОП.			
Итого	64	64	

Таблица 3.2

Таблица 3.3

Темы лабораторных занятий	Активные формы, час.	Часы	Ссылки на результаты обучения	Учебная деятельность
Семестр: 1				
Современные динамические языки	8	8	1,3,4,7,9,10	Решение практических задач
Современные функциональные языки	10	10	1,3,4,9,10	Решение практических задач
Элементы метапрограммирования, аспектно-ориентированное программирование.	14	14	1,2,3,4,5,7,8,9,10	Выполнение учебного проекта
Итого	32	32		

4. Самостоятельная работа студентов

Таблица 4.1

№	Виды самостоятельной работы	Ссылки на результаты обучения	Часы на выполнение	Часы на консультации
Семестр: 1				
1	изучение разделов дисциплины по учебной литературе, в том числе вопросов, не освещаемых на лекциях, решение практических задач, выполнение учебного проекта	1,2,3,4,5,7,8,9,10	38	
	Изучение предлагаемых теоретических разделов в соответствии с настоящей Программой. Учебно-методические материалы по дисциплине «Современные методы программирования» выложены на странице курса в сети Интернет			
2	Подготовка к занятиям, к текущему контролю знаний	1,2,3,4,5,7,8,9,10	90	
	Выполнение заданий			
3	Подготовка к экзамену	1,2,3,4,5,7,8,9,10	24	2
	Повторение теоретического материала по вопросам, совпадающим с темами лекций			
	Итого		152	2

5. Образовательные технологии

В ходе реализации учебного процесса по дисциплине применяются лекционные и лабораторные занятия, а также применяются следующие интерактивные формы обучения (таблица 5.1). Предусмотрено проведение лекций, лабораторных и практических занятий в дистанционном формате с использованием сервисов дистанционного взаимодействия (Google Meet). При проведении лабораторных и практических занятий студенты подключаются к онлайн сессии. На занятиях разбираются теоретические темы, представленные на лекции, и формулировки практических и лабораторных заданий. Для сдачи выполненного задания студент включает демонстрацию экрана, показывает результаты, обосновывает решение, отвечает на вопросы преподавателя.

Таблица 5.1

1	Лекция в форме дискуссии	ОПК-2.1, 2.2, 2.3, ОПК-5.2, 5.3, ОПК-6.1, 6.2, 6.3, ОПК-7.3, ОПК-8.1
Формируемые умения:		
Знать методы объектно-ориентированного, функционального, аспектно-ориентированного программирования. Уметь комбинировать различные языки и системы программирования, а также методы проектирования с целью оптимального решения поставленных задач проектирования ПО		
Краткое описание применения: Представляется теория, проблематика вопросов, связанных с проектированием пользовательских интерфейсов различного типа, обсуждаются особенности современных методов программирования, рассматриваются примеры, обсуждаются преимущества и недостатки.		
2	Портфолио	ОПК-2.1, 2.2, 2.3, ОПК-5.2, 5.3, ОПК-6.1, 6.2, 6.3, ОПК-7.3, ОПК-8.1

Формируемые умения:

Знать методы объектно-ориентированного, функционального, аспектно-ориентированного программирования. Уметь комбинировать различные языки и системы программирования, а также методы проектирования с целью оптимального решения поставленных задач проектирования ПО

Краткое описание применения: студенты ведут портфолио (оценки за задания), которое является основой для проведения аттестации по дисциплине

Для организации и контроля самостоятельной работы студентов, а также проведения консультаций применяются информационно-коммуникационные технологии (таблица 5.2).

Таблица 5.2

Информирование	Адрес почты – сообщается магистрантам на первом занятии,
Консультирование	Адрес почты – сообщается магистрантам на первом занятии.
Контроль	Адрес почты – сообщается магистрантам на первом занятии.
Размещение учебных материалов	Материалы размещаются в электронном ресурсе, создаваемом для каждого нового набора

6. Правила аттестации по учебной дисциплине

Текущая аттестация по дисциплине «Современные методы программирования» проводится в форме портфолио. Промежуточная аттестация проводится в формате экзамена.

В рамках самостоятельной работы обучающийся выполняет задания, а также занимается изучением необходимых для этого языков программирования (на лекциях даются только основы).

По первому и второму разделам курса текущий контроль осуществляется на основе выполненных студентом основных заданий. Задание должно быть сдано в течении 2-х учебных недель после того, как его условие сформулировано на лекции. Допускается сдача с дополнительной задержкой в 1 или 2 недели со штрафом к оценке в 1 или 2 балла соответственно. Далее задача считается не выполненной. Допускается сдача одной такой задачи в произвольное время без штрафа к оценке. Т.е. при условии выполнения всех задач во время одну задачу студент имеет право отложить на любое время (до получения допуска к экзамену) без штрафов. Сроки сдачи заданий по первому разделу (без штрафов) – 5-ая неделя, по второму – 10-ая.

Выполнение всех обязательных заданий является необходимым и достаточным условием для получения допуска к экзамену с предварительной оценкой «удовлетворительно».

Если студент желает повысить предварительную оценку до «хорошо» или «отлично» он должен полностью или частично решить одну дополнительную задачу, условие которой обсуждается индивидуально с преподавателем.

Пример задания

1. Задан набор символов и число n . Опишите функцию, которая возвращает список всех строк длины n , состоящих из этих символов и не содержащих двух одинаковых символов, идущих подряд. Например, для символов 'a', 'b', 'c' и $n=2$ результат должен

быть ("ab" "ac" "ba" "bc" "ca" "cb") с точностью до перестановки. Не допускается использовать циклы или рекурсию.

- По аналогии с задачей дифференцирования реализовать представление символьных булевых выражений с операциями конъюнкции, дизъюнкции отрицания, импликации. Выражения могут включать как булевы константы, так и переменные. Реализовать подстановку значения переменной в выражение с его приведением к ДНФ.

Код должен быть покрыт тестами, API документирован.

Примеры вопросов к экзамену:

- Интерпретация и компиляция. АОТ-, JIT- компиляция. Компиляция в динамических языках.
- Виды типизации. Преимущества и недостатки различных видов типизации. Типизация в динамических языках.
- Семантика регулярных выражений. Основные операции с регулярными выражениями.

Промежуточная аттестация:

Промежуточная аттестация по курсу проводится по результатам сдачи устного экзамена. Условием допуска к экзамену является получение положительной предварительной оценки (по результатам выполнения заданий).

Оценка за экзамен в свою очередь складывается из предварительной оценки и ответов на экзаменационные вопросы. При этом предварительная оценка может быть улучшена или ухудшена не более чем на 1 балл. Таким образом, студент имеет право отказаться отвечать на экзаменационные вопросы и получить оценки на балл ниже предварительной.

Промежуточная аттестация по дисциплине производится: в 1 семестре в виде экзамена.

По результатам аттестации выставляется оценка по шкале «неудовлетворительно», «удовлетворительно», «хорошо», «отлично». Оценки «отлично», «хорошо», «удовлетворительно» означают успешное прохождение промежуточной аттестации.

В таблице 6.1 представлено соответствие форм аттестации заявляемым требованиям к результатам освоения дисциплины.

Таблица 6.1

Коды компетенций ФГОС	Результаты обучения	Формы аттестации	
		семестр 1	
		портфолио	экзамен
ОПК-2	ОПК-2.1. Знать: современные информационно-коммуникационные и интеллектуальные технологии, инструментальные среды, программно-технические платформы для решения профессиональных задач	+	+
ОПК-2	ОПК-2.2 Уметь: обосновывать выбор современных информационно-коммуникационных и интеллектуальных технологий, разрабатывать оригинальные программные средства	+	+

	для решения профессиональных задач		
ОПК-2	ОПК-2.3. Владеть: методами разработки оригинальных программных средств, в том числе с использованием современных информационно-коммуникационных и интеллектуальных технологий, для решения профессиональных задач	+	+
ОПК-5	ОПК-5.2. Уметь: модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем для решения профессиональных задач	+	+
ОПК-5	ОПК-5.3. Владеть: навыками разработки программного и аппаратного обеспечения информационных и автоматизированных систем для решения профессиональных задач	+	+
ОПК-6	ОПК-6.1. Знать: аппаратные средства и платформы инфраструктуры информационных технологий, виды, назначение, архитектуру, методы разработки и администрирования программно-аппаратных комплексов объекта профессиональной деятельности	+	+
ОПК-6	ОПК-6.2. Уметь: анализировать техническое задание, разрабатывать и оптимизировать программный код для решения задач обработки информации и автоматизированного проектирования	+	+
ОПК-6	ОПК-6.3 Владеть: навыками составления технической документации по использованию и настройке компонентов программно-аппаратного комплекса	+	+
ОПК-7	ОПК-7.3. Владеть: навыками настройки интерфейса, разработки пользовательских шаблонов, подключения библиотек, добавления новых функций	+	+
ОПК-8	ОПК-8.1. Знать: методы и средства разработки программного обеспечения, методы управления проектами разработки программного обеспечения, способы организации проектных данных, нормативно-технические документы (стандарты и регламенты) по разработке программных средств и проектов	+	+

Требования к структуре и содержанию портфолио, оценочные средства, а также критерии оценки сформированности компетенций и освоения дисциплины в целом, представлены в Фонде оценочных средств, являющемся приложением 1 к настоящей рабочей программе дисциплины.

7. Литература

Основная литература

1. Фултон, Х. Программирование на языке Ruby : руководство / Х. Фултон. - Москва : ДМК Пресс, 2007. - 688 с. : ил. - ISBN 5-94074-357-9 ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=47356>
2. Городня Л.В.. Парадигмы программирования: анализ и сравнение. Из-во СО РАН РФФИ 17-11-00042. 216 с. http://www.rfbr.ru/rffi/ru/books/o_2043045
3. Городня Л.В.. Парадигма программирования (курс лекций) Новосиб. гос. Ун-т. – Новосибирск : РИЦ НГУ, 2015. – 206 с. <https://www.iis.nsk.su/files/book/file/FIT-Gor-PP3.pdf>
4. Городня Л.В. Парадигмы программирования. – М.: Интернет-Университет Информационных технологий. – <http://www.intuit.ru>, 2006.

Дополнительная литература:

5. Городня Л.В. Основы функционального программирования. – М.: Интернет-Университет Информационных технологий. – <http://www.intuit.ru>, 2004. – 272 с.

Интернет-ресурсы

Таблица 7.1

№ п/п	Наименование Интернет-ресурса	Краткое описание
1.	http://www.nsu.ru/xmlui/	Электронная библиотека НГУ
2.	http://www.spsl.nsc.ru	Портал ГПНТБ СО РАН
3.	http://libra.nsu.ru	НГУ. Научная электронная библиотека
4.	http://e.lanbook.com/	Электронно-библиотечная система «Лань»

8. Учебно-методическое и программное обеспечение дисциплины

8.1. Учебно-методическое обеспечение

Электронный курс выложен на платформе электронных учебных курсов в среде Интернет Мигинский Д.С. Современные методы программирования. Электронный курс лекций, НГУ, 2013 г. <https://classroom.google.com/u/1/c/MzcyNzlyMjY1NDIa>

8.2. Программное обеспечение

Для обеспечения реализации дисциплины используется стандартный комплект программного обеспечения (ПО), включающий регулярно обновляемое лицензионное ПО Windows и MS Office.

Перечень специализированного программного обеспечения для изучения дисциплины представлен в таблице 8.1.

Специализированное программное обеспечение Таблица 8.1

№	Наименование ПО	Назначение	Место размещения
1	MS Office (Word, Excel, PowerPoint) или аналоги	Программное обеспечение для демонстрации слайд-презентаций; текстовый, табличный редактор.	Аудитории 4220, 4218, 4214, 4213, 4211, 4210, 3220, 3218, 3213, 3212, 2221, 2213 Учебного корпуса №1
2	Eclipse	Среда разработки приложений	Аудитории 4220, 4218, 4214, 4213, 4211, 4210, 3220, 3218, 3213, 3212, 2221, 2213 Учебного корпуса №1

ПО для лиц с ограниченными возможностями здоровья Таблица 8.2

№	Наименование ПО	Назначение	Место размещения
1	Jaws for Windows	Программа экранного доступа к системным и офисным приложениям, включая интернет-обозреватели. Информация с экрана считывается вслух, обеспечивая возможность	Ресурсный центр, читальные залы библиотеки НГУ, компьютерные

		речевого доступа к самому разнообразному контенту. Jaws также позволяет выводить информацию на обновляемый дисплей Брайля. JAWS включает большой набор клавиатурных команд, позволяющих воспроизвести действия, которые обычно выполняются только при помощи мыши.	классы (сетевые лицензии)
2	Duxbury Braille Translator v11.3 для Брайлевского принтера	Программа перевода текста в текст Брайля, и печати на Брайлевском принтере	Ресурсный центр
3	"MAGic Pro 13" (увеличение+речь)	Программа для людей со слабым зрением и для незрячих людей. Программа позволяет увеличить изображение на экране до 36 крат, есть функция речевого сопровождения	Ресурсный центр, читальные залы библиотеки НГУ

9. Профессиональные базы данных и информационные справочные системы

1. Полнотекстовые журналы Springer Journals за 1997-2015 г., электронные книги (2005-2016 гг.), коллекция научных биомедицинских и биологических протоколов SpringerProtocols, коллекция научных материалов в области физических наук и инжиниринга SpringerMaterials, реферативная БД по чистой и прикладной математике zbMATH.

2. Электронная библиотека диссертаций Российской государственной библиотеки (ЭБД РГБ)

3. Электронные ресурсы Web of Science Core Collection (Thomson Reuters Scientific LLC.), Journal Citation Reports + ESI

4. БД Scopus (Elsevier)

10. Материально-техническое обеспечение

Таблица 10.1

№	Наименование	Назначение
1	Презентационное оборудование (мультимедиа-проектор, экран, компьютер для управления)	Для проведения лекционных и семинарских занятий
2	Компьютерный класс (с выходом в Internet)	Для организации самостоятельной работы обучающихся

Оборудование, обеспечивающее адаптацию электронных и печатных образовательных ресурсов для обучающихся из числа лиц с ограниченными возможностями здоровья

Таблица 10.2

№	Наименование оборудования	Назначение	Место размещения
1	Принтер Брайля	Печать рельефно-точечным шрифтом Брайля	Ресурсный центр
2	Увеличитель Prodigy Duo Tablet 24	Устройство для чтения и увеличения плоскочечного текста	Ресурсный центр, читальные залы библиотеки НГУ

3	Специализированное мобильное рабочее место «ЭлНот 311»	Мобильный компьютер с дисплеем брайля	Ресурсный центр
4	Портативный тактильный дисплей Брайля “Focus 40 Blue”	Навигация в операционных системах, программах и интернете с помощью отображения рельефно-точечным шрифтом Брайля получаемой информации	Ресурсный центр, читальные залы библиотеки НГУ
5	Устройство для печати тактильной графики «PIAF»	Печать тактильных графических изображений	Ресурсный центр
6	Портативный видео-увеличитель RUBY XL HD	Увеличение текста и подбор контрастных схем изображения	Ресурсный центр
7	Складной настольный электронный видео-увеличитель «ТОPAZ PHD 15»	Увеличение текста и подбор контрастных схем изображения	Ресурсный центр
8	Электронный ручной видео-увеличитель ONYX Deskset HD 22”	Увеличение текста и подбор контрастных схем изображения	Ресурсный центр
9	Смартфон EISmart G3	Смартфон клавишным управлением и озвученным интерфейсом, обучение спутниковой навигации.	Ресурсный центр
10	FM-система «Сонет-PCM» PM-3-1	Звуковая FM-система для людей с нарушением слуха, улучшающая восприятие голосовой информации	Большая физическая аудитория главного корпуса НГУ

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования «Новосибирский национальный исследовательский
государственный университет» (Новосибирский государственный университет, НГУ)

Факультет информационных технологий

СОГЛАСОВАНО

Декан ФИТ НГУ

 М.М. Лаврентьев

«25» апреля 2023 г.

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ
по дисциплине Современные методы программирования**

Направление подготовки: 09.04.01 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА

Направленность (профиль): Искусственный интеллект и Data Science

Квалификация: магистр

Форма обучения: очная

Год обучения: 1, семестр 1

Форма аттестации	Семестр
Экзамен	1

Новосибирск 2023

Фонд оценочных средств промежуточной аттестации по дисциплине является **Приложением 1** к рабочей программе дисциплины «Современные методы программирования», реализуемой в рамках образовательной программы высшего образования – программы магистратуры 09.04.01 Информатика и вычислительная техника, направленность (профиль): Искусственный интеллект и Data Science

Фонд оценочных средств промежуточной аттестации по дисциплине утвержден решением ученого совета факультета информационных технологий, протокол №91 от 24.04.2023.

Разработчик:

Доцент кафедры систем информатики ФИТ,
кандидат физико-математических наук

 Д.С.Мигинский

Заведующий кафедрой систем информатики ФИТ,
доктор физико-математических наук

 М.М. Лаврентьев

Ответственный за образовательную программу:

Заведующий кафедрой систем информатики ФИТ,
доктор физико-математических наук

 М.М. Лаврентьев

1. Содержание и порядок проведения промежуточной аттестации по дисциплине

1.1. Общая характеристика содержания промежуточной аттестации

Промежуточная аттестация по дисциплине «Современные методы программирования» проводится по завершению периода освоения образовательной программы (семестра) для оценки сформированности компетенций в части следующих индикаторов достижения компетенции (таблица П1.1).

Таблица П1.1

Код	Компетенции, формируемые в рамках дисциплины «Современные методы программирования»	Семестр 1	
		Портфолио	Экзамен
	ОПК-2. Способен разрабатывать оригинальные алгоритмы и программные средства, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач		
ОПК-2.1.	Знать: современные информационно-коммуникационные и интеллектуальные технологии, инструментальные среды, программно-технические платформы для решения профессиональных задач	+	+
ОПК-2.2	Уметь: обосновывать выбор современных информационно-коммуникационных и интеллектуальных технологий, разрабатывать оригинальные программные средства для решения профессиональных задач	+	+
ОПК-2.3.	Владеть: методами разработки оригинальных программных средств, в том числе с использованием современных информационно-коммуникационных и интеллектуальных технологий, для решения профессиональных задач	+	+
	ОПК-5. Способен разрабатывать и модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем		
ОПК-5.2.	Уметь: модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем для решения профессиональных задач	+	+
ОПК-5.3.	Владеть: навыками разработки программного и аппаратного обеспечения информационных и автоматизированных систем для решения профессиональных задач	+	+
	ОПК-6. Способен разрабатывать компоненты программно-аппаратных комплексов обработки информации и автоматизированного проектирования		
ОПК-6.1.	Знать: аппаратные средства и платформы инфраструктуры информационных технологий, виды, назначение, архитектуру, методы разработки и администрирования программно-аппаратных комплексов объекта профессиональной деятель-	+	+

	ности		
ОПК-6.2.	Уметь: анализировать техническое задание, разрабатывать и оптимизировать программный код для решения задач обработки информации и автоматизированного проектирования	+	+
ОПК-6.3	Владеть: навыками составления технической документации по использованию и настройке компонентов программно-аппаратного комплекса	+	+
	ОПК-7. Способен адаптировать зарубежные комплексы обработки информации и автоматизированного проектирования к нуждам отечественных предприятий		
ОПК-7.3.	Владеть: навыками настройки интерфейса, разработки пользовательских шаблонов, подключения библиотек, добавления новых функций	+	+
	ОПК-8. Способен осуществлять эффективное управление разработкой программных средств и проектов		
ОПК-8.1.	Знать: методы и средства разработки программного обеспечения, методы управления проектами разработки программного обеспечения, способы организации проектных данных, нормативно-технические документы (стандарты и регламенты) по разработке программных средств и проектов	+	+

Тематика вопросов к экзамену соответствует избранным разделам (темам) дисциплины «Современные методы программирования»

Раздел 1: Современные динамические языки

Раздел 2. Современные функциональные языки

Раздел 3. Элементы метапрограммирования, аспектно-ориентированное программирование.

1.2. Порядок проведения промежуточной аттестации по дисциплине

Экзамен проводится в устной форме, в аудитории, студентам разрешено пользоваться бумагой для записей и авторучкой. Справочной, учебной и другой литературой пользоваться не разрешается. Использование электронных устройств (телефоны, любые виды компьютеров, т.д.) запрещено.

2. Требования к структуре и содержанию фонда оценочных средств промежуточной аттестации по дисциплине

Перечень оценочных средств, применяемых на каждом этапе проведения промежуточной аттестации по дисциплине, представлен в таблице П1.2.

Таблица П1.2

№ п/п	Наименование оценочного средства	Краткая характеристика оценочного средства	Представление оценочного средства в фонде
Этап 1 - портфолио			
1.	Портфолио	Целевая подборка работ студента, раскрывающая его индивидуальные обра-	Структура портфолио

		зовательные достижения в одной или нескольких учебных дисциплинах.	
Этап 2 – Экзамен			
2	Экзаменационный билет	Комплекс вопросов	Список теоретических вопросов

2.1. Требования к структуре и содержанию оценочных средств аттестации в 1 семестре

Текущая аттестация по дисциплине «Современные методы программирования» проводится в форме портфолио. Промежуточная аттестация проводится в формате экзамена.

Состав портфолио:

В рамках самостоятельной работы обучающийся выполняет задания, входящие в рамки портфолио.

По первому и второму разделам курса текущий контроль осуществляется на основе выполненных студентом основных заданий. Задание должно быть сдано в течении 2-х учебных недель после того, как его условие сформулировано на лекции. Допускается сдача с дополнительной задержкой в 1 или 2 недели со штрафом к оценке в 1 или 2 балла соответственно. Далее задача считается не выполненной. Допускается сдача одной такой задачи в произвольное время без штрафа к оценке. Т.е. при условии выполнения всех задач во время одну задачу студент имеет право отложить на любое время (до получения допуска к экзамену) без штрафов. Таким образом, сроки сдачи заданий по первому разделу (без штрафов) – 6-ая неделя, по второму – 12-ая.

Выполнение всех обязательных заданий является необходимым и достаточным условием для получения допуска к экзамену с предварительной оценкой «удовлетворительно».

Если студент желает повысить предварительную оценку до «хорошо» или «отлично» он должен полностью или частично решить одну дополнительную задачу, условие которой обсуждается индивидуально с преподавателем.

Пример основного задания

По аналогии с задачей дифференцирования реализовать представление символьных булевых выражений с операциями конъюнкции, дизъюнкции отрицания, импликации. Выражения могут включать как булевы константы, так и переменные. Реализовать подстановку значения переменной в выражение с его приведением к ДНФ. Код должен быть покрыт тестами, API документирован.

Пример дополнительного задания

Представление данных в виде S-выражений.

- 1) Определите формат представления данных с древовидной структурой в виде S-выражений. В качестве семантической основы рекомендуется рассмотреть форматы XML и/или JSON. Допускается использование как классических S-выражений, так и расширенных, используемых в языке Clojure.
- 2) Разработайте язык навигации и простых запросов для заданного формата. Должны поддерживаться: относительный или абсолютный путь к узлу дерева, путь с условиями на свойства промежуточных узлов, путь с переменной вложенностью. В качестве семантической основы рекомендуется рассмотреть язык XPath для XML. Реализуйте функции поиска и модификации заданного формата посредством этого языка.

Дополнительно.

- 1) Разработайте представление схемы (по аналогии с XML Schema). Реализуйте функции проверки документа по схеме.
- 2) Разработайте язык трансформации в HTML, как упрощенный аналог XSLT.

Примеры вопросов к экзамену:

Основные характеристики и области применения динамических языков.

1. Интерпретация и компиляция. AOT-, JIT- компиляция. Компиляция в динамических языках.
2. Виды типизации. Преимущества и недостатки различных видов типизации. Типизация в динамических языках.
3. Язык Ruby, классификация, основные реализации. Встроенные структуры данных.
4. Семантика регулярных выражений. Основные операции с регулярными выражениями.
5. Объектная модель Ruby.
6. Динамическое изменение объектной модели в Ruby: элементы Meta-Object Protocol.
7. Основные положения функциональной парадигмы программирования. Неподвижное состояние, преимущества и недостатки.
8. Понятие лексического контекста. Замыкания.
9. Побочные эффекты. Чистые функции. Преимущества и недостатки.
10. Функции как объекты первого класса. Функции высших порядков: функционал, оператор. Примеры. Операторы композиции и частичного применения (каррирования).
11. Основные функции преобразования коллекций: map, reduce, filter. Отличия от прямой итерации по коллекции. Примеры использования (на Ruby или Clojure).
12. Язык Clojure. Концепция LISP: код как данные, связь с АСД. Модель компиляции/исполнения. REPL.
13. Встроенные типы и структуры данных Clojure. Основные операции.
14. Основные управляющие структуры Clojure: вызов функции, ветвления, цикл. Связь рекурсии и цикла. Хвостовая рекурсия в Clojure. Императивные управляющие структуры: последовательное исполнение, doseq.
15. Генератор for в Clojure, связь с map/reduce/filter.
16. Мемоизация. Область и примеры применения.
17. Реорганизующее присваивание (destructuring) в Clojure. Использование при объявлении/вызове функции, в let.
18. Отложенные вычисления на примере Clojure. Ленивые последовательности, delay. Основные операции над ленивыми последовательностями.
19. Потoki данных. Моделирование состояния с помощью потоков. Примеры использования. Бесконечные потоки.
20. Квотирование (quote). Виды квотирования в Clojure. Обратные операции: unquote, eval.
21. Специальные формы. Макросы. Модель исполнения. Применение макросов.
22. Особенности разрешения символов в eval и макросах. Внутренние переменные в макросах.
23. Разделение ответственностей. Принцип KISS. Связь с модульностью и абстракцией.

24. Ответственности 2-го класса (cross-cutting concerns). Примеры. Способы разделения ответственностей.
25. Инверсия управления (принцип Голливуда). Примеры применения.
26. Внедрение зависимостей (Dependency Injection, DI). Элементарное DI. DI с использованием контейнера. Связь с порождающими шаблонами проектирования.
27. Формы управления параллелизмом без блокировок.
28. Atomic-типы. Atomic-ссылка в Clojure, основные операции. Агенты. Future, promise.
29. Транзакционная память. Multi-Version Concurrency Control. Реализация в Clojure: алгоритм выполнения транзакции, изоляция транзакций.
30. Понятие распределенной транзакции. CAP-теорема. Транзакции типа Copy-Modify-Merge.
31. Формы полиморфизма. Полиморфизм в динамических языках. Принцип подстановки Барбары Лисков (строгая формулировка). Интерпретация в контрактном программировании.
32. Полиморфизм в иерархиях с одиночным и множественным наследованием. Комбинация методов на примере CLOS.
33. Обобщенные функции и посылка сообщений. Диспетчеризация по нескольким параметрам. Примеры использования.
34. Вспомогательные методы. Комбинация в иерархии наследования на примере CLOS. Примеры использования для разделения ответственностей.
35. Command-Query Separation. Применение в функциональных и императивных объектных моделях.
36. Аспектно-ориентированное программирование. Перехват, способы реализации. Применение для разделения ответственностей.
37. Понятие динамического лексического контекста. Реализация в Clojure. Связь с АОП. Применение для разделения ответственностей.

Промежуточная аттестация:

Промежуточная аттестация по курсу проводится по результатам сдачи устного экзамена. Для получения положительной оценки на экзамене необходимо сдать все обязательные задания (предварительная оценка)

Оценка за экзамен в свою очередь складывается из предварительной оценки и ответов на экзаменационные вопросы.

Экзаменационный билет включает два теоретических вопроса по разным темам курса.

При этом предварительная оценка может быть улучшена или ухудшена не более чем на 1 балл. Таким образом, студент имеет право отказаться отвечать на экзаменационные вопросы и получить оценки на балл ниже предварительной.

Промежуточная аттестация по дисциплине производится: в 1 семестре в виде экзамена.

По результатам аттестации выставляется оценка по шкале «неудовлетворительно», «удовлетворительно», «хорошо», «отлично». Оценки «отлично», «хорошо», «удовлетворительно» означают успешное прохождение промежуточной аттестации.

2.2. Форма и перечень вопросов экзаменационного билета 1 семестра

Форма экзаменационного билета

Таблица П1.3

<p>Новосибирский государственный университет Экзамен</p> <p>Современные методы программирования наименование дисциплины</p> <p>09.04.01 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА. Технология разработки программных систем наименование образовательной программы</p> <p>ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №</p> <p>1. Вопрос 1 2. Вопрос 2</p> <p>Составитель _____ Д.С.Мигинский</p> <p>Ответственный за образовательную программу _____ М.М.Лаврентьев (подпись)</p> <p>« ____ » _____ 20 ____ г.</p>

Перечень вопросов экзамена, структурированный по категориям, представлен в таблице П1.4

Таблица П1.4

Категория	Формулировка вопроса
Категория 1	1. Основные характеристики и области применения динамических языков.
	2. Интерпретация и компиляция. АОТ-, JIT- компиляция. Компиляция в динамических языках.
	3. Виды типизации. Преимущества и недостатки различных видов типизации. Типизация в динамических языках.
	4. Язык Ruby, классификация, основные реализации. Встроенные структуры данных.
	5. Семантика регулярных выражений. Основные операции с регулярными выражениями.

	6. Объектная модель Ruby.
	7. Динамическое изменение объектной модели в Ruby: элементы Meta-Object Protocol.
	8. Основные положения функциональной парадигмы программирования. Неподвижное состояние, преимущества и недостатки.
	9. Понятие лексического контекста. Замыкания.
	10. Побочные эффекты. Чистые функции. Преимущества и недостатки.
	11. Функции как объекты первого класса. Функции высших порядков: функционал, оператор. Примеры. Операторы композиции и частичного применения (каррирования).
	12. Основные функции преобразования коллекций: map, reduce, filter. Отличия от прямой итерации по коллекции. Примеры использования (на Ruby или Clojure).
	13. Язык Clojure. Концепция LISP: код как данные, связь с АСД. Модель компиляции/исполнения. REPL.
	14. Встроенные типы и структуры данных Clojure. Основные операции.
	15. Основные управляющие структуры Clojure: вызов функции, ветвления, цикл. Связь рекурсии и цикла. Хвостовая рекурсия в Clojure. Императивные управляющие структуры: последовательное исполнение, doseq.
	16. Генератор for в Clojure, связь с map/reduce/filter.
	17. Мемоизация. Область и примеры применения.
	18. Реорганизуемое присваивание (destructuring) в Clojure. Использование при объявлении/вызове функции, в let.
	19. Отложенные вычисления на примере Clojure. Ленивые последовательности, delay. Основные операции над ленивыми последовательностями.
Категория 2	20. Потoki данных. Моделирование состояния с помощью потоков. Примеры использования. Бесконечные потоки.
	21. Квотирование (quote). Виды квотирования в Clojure. Обратные операции: unquote, eval.
	22. Специальные формы. Макросы. Модель исполнения. Приме-

	нение макросов.
	23. Особенности разрешения символов в eval и макросах. Внутренние переменные в макросах.
	24. Разделение ответственностей. Принцип KISS. Связь с модульностью и абстракцией.
	25. Ответственности 2-го класса (cross-cutting concerns). Примеры. Способы разделения ответственностей.
	26. Инверсия управления (принцип Голливуда). Примеры применения.
	27. Внедрение зависимостей (Dependency Injection, DI). Элементарное DI. DI с использованием контейнера. Связь с порождающими шаблонами проектирования.
	28. Формы управления параллелизмом без блокировок.
	29. Atomic-типы. Atomic-ссылка в Clojure, основные операции. Агенты. Future, promise.
	30. Транзакционная память. Multi-Version Concurrency Control. Реализация в Clojure: алгоритм выполнения транзакции, изоляция транзакций.
	31. Понятие распределенной транзакции. CAP-теорема. Транзакции типа Copy-Modify-Merge.
	32. Формы полиморфизма. Полиморфизм в динамических языках. Принцип подстановки Барбары Лисков (строгая формулировка). Интерпретация в контрактном программировании.
	33. Полиморфизм в иерархиях с одиночным и множественным наследованием. Комбинация методов на примере CLOS.
	34. Обобщенные функции и посылка сообщений. Диспетчеризация по нескольким параметрам. Примеры использования.
	35. Вспомогательные методы. Комбинация в иерархии наследования на примере CLOS. Примеры использования для разделения ответственностей.
	36. Command-Query Separation. Применение в функциональных и императивных объектных моделях.
	37. Аспектно-ориентированное программирование. Перехват, способы реализации. Применение для разделения ответственностей.

	38. Понятие динамического лексического контекста. Реализация в Clojure. Связь с АОП. Применение для разделения ответственностей.
--	--

Набор экзаменационных билетов формируется и утверждается в установленном порядке в начале учебного года при наличии контингента обучающихся, завершающих освоение дисциплины «Современные методы программирования» в текущем учебном году.

3. Критерии оценки сформированности компетенций в рамках промежуточной аттестации по дисциплине

Таблица П1.5

Шифр компетенций	Структурные элементы оценочных средств	Показатель сформированности	Не сформирован	Пороговый уровень	Базовый уровень	Продвинутый уровень
ОПК-2	Портфолио (этап 1), Экзамен (этап 2)	ОПК-2.1. Знать: современные информационно-коммуникационные и интеллектуальные технологии, инструментальные среды, программно-технические платформы для решения профессиональных задач ОПК-2.2 Уметь: обосновывать выбор современных информационно-коммуникационных и интеллектуальных технологий, разрабатывать оригинальные программные средства для решения профессиональных задач ОПК-2.3. Владеть: методами разработки оригинальных программных средств, в том числе с использованием современных информационно-коммуникационных и интеллектуальных технологий, для решения профессиональных задач	Не знает современные информационно-коммуникационные и интеллектуальные технологии, инструментальные среды, программно-технические платформы для решения профессиональных задач	Демонстрирует фрагментарные знания современных инструментальных сред, программно-технические платформы для решения профессиональных задач	Демонстрирует незначительные погрешности, знает основные современные информационно-коммуникационные и интеллектуальные технологии, инструментальные среды, программно-технические платформы для решения профессиональных задач	Демонстрирует целостное знание, хорошо ориентируется в современных информационно-коммуникационных и интеллектуальных технологиях, инструментальных средах, программно-технических платформах для решения профессиональных задач
ОПК-5	Портфолио (этап 1), Экзамен (этап 2)	ОПК-5.2. Уметь: модернизировать программное и	Не умеет разрабаты-	Допускает грубые ошибки	Умеет разрабатывать программное	Умеет уверенно разрабатывать программное

		<p>аппаратное обеспечение информационных и автоматизированных систем для решения профессиональных задач</p> <p>ОПК-5.3. Владеть: навыками разработки программного и аппаратного обеспечения информационных и автоматизированных систем для решения профессиональных задач</p>	<p>вать программное и аппаратное обеспечение информационных и автоматизированных систем для решения профессиональных задач</p>	<p>при разработке программного и аппаратного обеспечения информационных и автоматизированных систем для решения профессиональных задач</p>	<p>и аппаратное обеспечение информационных и автоматизированных систем для решения учебно-профессиональных задач</p>	<p>и аппаратное обеспечение информационных и автоматизированных систем для решения широкого класса профессиональных задач</p>
ОПК-6	<p>Портфолио (этап 1), Экзамен (этап 2)</p>	<p>ОПК-6.1. Знать: аппаратные средства и платформы инфраструктуры информационных технологий, виды, назначение, архитектуру, методы разработки и администрирования программно-аппаратных комплексов объекта профессиональной деятельности</p> <p>ОПК-6.2. Уметь: анализировать техническое задание, разрабатывать и оптимизировать программный код для решения задач обработки информации и автоматизированного проектирования</p> <p>ОПК-6.3 Владеть: навы-</p>	<p>Не знает: аппаратные средства и платформы инфраструктуры информационных технологий, виды, назначение, архитектуру, методы разработки и администрирования про-</p>	<p>Демонстрирует фрагментарное знание аппаратных средств и платформ инфраструктуры информационных технологий</p>	<p>Допускает незначительные погрешности, знает основы аппаратных средств и платформ инфраструктуры информационных технологий</p>	<p>Демонстрирует уверенное знание аппаратных средств и платформ инфраструктуры информационных технологий, видов, назначений, архитектур, методов разработки и администрирования программно-аппаратных комплексов объекта профессиональной деятельности</p>

		ками составления технической документации по использованию и настройке компонентов программно-аппаратного комплекса	граммно-аппаратных комплексов объекта профессиональной деятельности			
ОПК-7	Портфолио (этап 1), Экзамен (этап 2)	ОПК-7.3. Владеть: навыками настройки интерфейса, разработки пользовательских шаблонов, подключения библиотек, добавления новых функций	Не умеет подключать библиотеки, добавлять новые функции	Допускает грубые ошибки при подключении библиотеки, добавлении новых функций	Умеет подключать библиотеки, добавлять новые функции в рамках учебных задач	Умеет уверенно подключать библиотеки, добавлять новые функции для широкого спектра задач
ОПК-8	Портфолио (этап 1), Экзамен (этап 2)	ОПК-8.1. Знать: методы и средства разработки программного обеспечения, методы управления проектами разработки программного обеспечения, способы организации проектных данных, нормативно-технические документы (стандарты и регламенты) по разработке программных средств и проектов	Не знает методы и средства разработки программного обеспечения, методы управления проектами разработки программного обеспечения	Демонстрирует фрагментарные знания методов и средств разработки программного обеспечения, методов управления проектами разработки программного обеспечения	Демонстрирует незначительные погрешности в знании методов и средств разработки программного обеспечения, методов управления проектами разработки программного обеспечения	Демонстрирует целостное знание методов и средств разработки программного обеспечения, методов управления проектами разработки программного обеспечения

4. Критерии выставления оценок по результатам промежуточной аттестации по дисциплине

В соответствии с учебным планом устанавливаются следующие формы контроля:

В 1 семестре результаты промежуточной аттестации определяются оценками «отлично», «хорошо», «удовлетворительно», «неудовлетворительно». Оценки «отлично», «хорошо», «удовлетворительно» означают успешное прохождение промежуточной аттестации.

Оценка «отлично» соответствует продвинутому уровню сформированности компетенции.

Оценка «хорошо» соответствует базовому уровню сформированности компетенции.

Оценка «удовлетворительно» соответствует пороговому уровню сформированности компетенции.

Оценка «неудовлетворительно» выставляется, если хотя бы одна компетенция не сформирована.

Итоговая оценка результатов промежуточной аттестации выставляется как оценка за экзамен.