

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКАЯ ГОСУДАРСТВЕННАЯ АКАДЕМИЯ»

«УТВЕРЖДАЮ»

Проректор по учебной работе

« 27 » 03

2026 г.

Г.Ю. Нагорная



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Системы программирования

Уровень образовательной программы бакалавриат

Направление подготовки 01.03.02 Прикладная математика и информатика

Направленность (профиль) Прикладная математика и информатика

Форма обучения очная

Срок освоения ОП 4 года

Институт Цифровых технологий

Кафедра разработчик РПД Математика

Выпускающая кафедра Математика

Начальник
учебно-методического управления

Семенова Л.У.

Директор института ЦТ

Кумратова А.М.

Заведующий выпускающей кафедрой

Кочкаров А.М.

г. Черкесск, 2026 г.

СОДЕРЖАНИЕ

1. Цели освоения дисциплины	4
2. Место дисциплины в структуре образовательной программы	5
3. Планируемые результаты обучения по дисциплине	6
4. Структура и содержание дисциплины	7
4.1. Объем дисциплины и виды работы.....	7
4.2. Содержание дисциплины	8
4.2.1. Разделы (темы) дисциплины, виды учебной деятельности и формы контроля.....	8
4.2.2. Лекционный курс.....	9
4.2.3. Лабораторный практикум	10
4.2.4. Практические занятия	12
4.3. Самостоятельная работа обучающегося.....	12
5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине	13
6. Образовательные технологии	16
7. Учебно-методическое и информационное обеспечение дисциплины	17
7.1. Перечень основной и дополнительной учебной литературы.....	17
7.2. Перечень ресурсов информационно-телекоммуникационной сети «Интернет».....	17
7.3. Информационные технологии, лицензионное программное обеспечение	18
8. Материально-техническое обеспечение дисциплины	19
8.1. Требования к аудиториям (помещениям, местам) для проведения занятий.....	19
8.2. Требования к оборудованию рабочих мест преподавателя и обучающихся.....	19
8.3. Требования к специализированному оборудованию.....	19
9. Особенности реализации дисциплины для инвалидов и лиц с ограниченными возможностями здоровья	20
Приложение 1. Фонд оценочных средств	21
Приложение 2. Аннотация дисциплины	32
Рецензия на рабочую программу	33
Лист переутверждения рабочей программы дисциплины	34

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Целью освоения дисциплины «Системы программирования» является ознакомление обучающихся с основными инструментариями, предназначенными для работы с программами на одном из языков программирования, разработка системного и прикладного программно обеспечения осуществляется с помощью систем программирования, в состав которых входят: трансляторы с языков высокого уровня; средства редактирования, компоновки и загрузки программ; макроассемблеры (машинно-ориентированные языки);

Задачи дисциплины:

- освоение обучающимися основных принципов и методов системы программирования;
- изучение средств редактирования, компоновки и загрузки программ;
- изучение программно-аппаратных методов и средств системы программирования;
- изучение машинно-ориентированных языков, в частности ассемблер;
- осуществлять разработку кода программного модуля на современных языках программирования;
- создавать программу по разработанному алгоритму как отдельный модуль;
- выполнять отладку и тестирование программы на уровне модуля

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

2.1. Дисциплина «Системы программирования» относится к части, формируемой участниками образовательных отношений Блока 1. Дисциплины (модули) выбираемые обучающимися, имеет тесную связь с другими дисциплинами.

2.2. Ниже приведены предшествующие и последующие дисциплины, направленные на формирование компетенций дисциплины в соответствии с матрицей компетенций ОП.

Предшествующие и последующие дисциплины, направленные на формирование компетенций

№ п/п	Предшествующие дисциплины	Последующие дисциплины
1	Основы информатики Языки и методы программирования Основы управления и программирования БАС	Технологии программирования в Internet Компьютерная графика Криптография Выполнение и защита выпускной квалификационной работы

3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Планируемые результаты освоения образовательной программы (ОП) – компетенции обучающихся определяются требованиями стандарта по направлению подготовки 01.03.02 Прикладная математика и информатика и формируются в соответствии с матрицей компетенций ОП

№ п/п	Номер/ индекс компетенции	Наименование компетенции (или ее части)	В результате изучения дисциплины обучающиеся должны:
1	2	3	4
1.	ПК-3	Способен осуществлять программную реализацию алгоритмов при разработке прикладных информационных систем	ПК-3.1 Обладает знаниями в области программирования и современных компьютерных технологий при разработке прикладных информационных систем. Способен осуществлять оптимизацию алгоритмов при создании прикладных программ или прикладных комплексов ПК-3.2 Анализирует стандартные технические требования к проектированию программного обеспечения. ПК-3.3 Способен осуществлять разработку технических требований к программному обеспечению, при его создании

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

4.1. Объем дисциплины и виды работы

Очная форма обучения

Вид учебной работы		Семестр	
		Всего часов	№4, часов
1		2	3
Аудиторная контактная работа (всего)		54	54
В том числе:			
Лекции (Л)		18	18
Лабораторные работы (ЛР)		36	36
Контактная внеаудиторная работа, в том числе		1,7	1,7
Индивидуальные и групповые консультации		1,7	1,7
Самостоятельная работа обучающегося (СРО) (всего)		52	52
<i>Подготовка к лабораторным работам (ЛР)</i>		20	20
<i>Работа с книжными и электронными источниками</i>		10	10
<i>Подготовка к промежуточному тестовому контролю</i>		22	22
Промежуточная аттестация	Зачет, в том числе	3	3
	Прием зачета, час	0,3	0,3
ИТОГО: Общая трудоемкость	часов	108	108
	зач. ед.	3	3

4.2. Содержание дисциплины

4.2.1. Разделы (темы) дисциплины, виды учебной деятельности и формы контроля

Очная форма обучения

№ п/п	Наименование раздела (темы) дисциплины	Виды учебной деятельности, включая самостоятельную работу обучающихся (в часах)					Формы текущей и промежуточной аттестации
		Л	ЛР	ПЗ	СРО	все го	
1	2	3	4	5	6	7	8
Семестр 4							
1.	Раздел 1. Системы программирования для разработки программ на Java.	6	6		10	22	контрольные вопросы
2.	Раздел 2. Компиляторы и интерпретаторы. Интегрированная среда разработки программ	4	10		20	34	контрольные вопросы
3.	Раздел 3. Библиотеки стандартных классов и функций. Инструменты отладки	4	10		20	34	контрольные вопросы
4.	Раздел 4. Графические библиотеки и работа с утилитами	4	10		2	16	контрольные вопросы
5	Контактная внеаудиторная работа, в том числе индивидуальные и групповые консультации					1,7	индивидуальные и групповые консультации
	Промежуточная аттестация					0,3	Зачет
ИТОГО часов в 4 семестре:		18	36		52	108	
Всего:		18	36		52	108	

4.2.2. Лекционный курс

№ п/п	Наименование раздела дисциплины	Наименование темы лекции	Содержание лекции	Всего часов
1	2	3	4	5
Семестр 4				
1.	Раздел 1. Системы программирования для разработки программ на Java	1.1 Системы программирования. Основные этапы решения задачи	Постановка задачи на естественном языке. Разработка математической модели. Выбор алгоритма и структуры данных. Реализация на языке Java. Основные этапы жизненного цикла ПО.	2
		1.2 Тестирование и отладка программы. Документация и инструкции пользователю	Отладка в IDE (ошибки компиляции и времени выполнения). Тестирование (юнит-тесты, JUnit). Документирование программ (JavaDoc). Разработка инструкций пользователю. Жизненный цикл программы в Java-экосистеме.	4
2.	Раздел 2. Компиляторы и интерпретаторы. Интегрированная среда разработки программ	2.1 Компиляторы и интерпретаторы	Архитектура Java: компиляция исходного кода в байткод (javac). Виртуальная машина Java (JVM) как интерпретатор и JIT-компилятор. Особенности кроссплатформенности Java.	4
3.	Раздел 3. Библиотеки стандартных классов и функций. Инструменты отладки	3.1 Стандартная библиотека Java (Java Standard API)	Основные пакеты Java: java.lang, java.util, java.io, java.time и др. Использование библиотек для работы со строками, коллекциями, потоками ввода-	4

			вывода. Средства отладки в IDE (IntelliJ IDEA, Eclipse, VS Code).	
4.	Раздел 4. Графические библиотеки и работа с утилитами	4.1 Графические библиотеки и средства для работы с библиотеками	Обзор графических библиотек Java: AWT, Swing, JavaFX. Использование утилит для управления библиотеками (Maven, Gradle). Подключение внешних библиотек в проект.	4
ИТОГО часов в 4 семестре:				18
Всего:				18

4.2.3. Лабораторный практикум

№ п/п	Наименование раздела дисциплины	Наименование лабораторной работы	Содержание лабораторной работы	Всего часов
1	2	3	4	5
Семестр 4				
1.	Раздел 1. Системы программирования для разработки программ на Java	1.1 Изучение среды программирования. Основные этапы решения задачи	Постановка задачи. Разработка алгоритма. Создание простейшей программы на Java («Hello, World!»). Компиляция и запуск через javac/java и через IDE (IntelliJ IDEA, Eclipse, VS Code).	2
		1.2 Тестирование и отладка программы	Использование отладчика в IDE (точки останова, пошаговое выполнение, просмотр переменных). Разработка тестов с использованием JUnit. Сравнение ожидаемых и полученных результатов.	2
		1.3 Документирование программы. Инструкции пользователю	Написание комментариев и JavaDoc. Создание HTML-	2

			документации. Разработка краткой инструкции для пользователя программы. Обсуждение жизненного цикла ПО.	
2.	Раздел 2. Компиляторы и интерпретаторы. Интегрированные среды разработки	2.1 Работа с компилятором и JVM	Написание и компиляция Java-программы. Анализ полученного байткода с помощью javap. Исследование принципа работы JVM (интерпретация и JIT-компиляция). Сравнение запуска через командную строку и IDE.	10
3.	Раздел 3. Библиотеки стандартных классов и функций. Отладочные инструменты	3.1 Использование стандартных библиотек Java	Написание программ с использованием пакетов java.lang, java.util, java.io, java.time. Решение задач на строки, коллекции (ArrayList, HashMap), ввод-вывод. Использование встроенных средств отладки IDE.	10
4.	Раздел 3. Библиотеки стандартных классов и функций. Отладочные инструменты	3.1 Использование стандартных библиотек Java	Написание программ с использованием пакетов java.lang, java.util, java.io, java.time. Решение задач на строки, коллекции (ArrayList, HashMap), ввод-вывод. Использование встроенных средств отладки IDE.	10
ИТОГО часов в 4 семестре:				36
Всего:				36

4.2.3. Практические занятия не предусмотрены

4.3. Самостоятельная работа обучающегося

№ п/п	Наименование раздела (темы) дисциплины	№ п/п	Виды СРО	Всего Часов
1	3	4	5	6
Семестр 4				
1.	Раздел 1. Системы программирования для разработки программ на Java.	1.1	Работа с основной и дополнительной литературой. Подготовка к лабораторному практикуму.	10
2.	Раздел 2. Компиляторы и интерпретаторы. Интегрированная среда разработки программ	2.1	Подготовка к лабораторному практикуму. Выполнения индивидуальных заданий по лабораторному практикуму	20
3.	Раздел 3. Библиотеки стандартных классов и функций. Инструменты отладки	3.1	Работа с книжными и электронными источниками. Изучение теоретического материала для выполнения лабораторных работ	20
4.	Раздел 4. Графические библиотеки и работа с утилитами	4.1	Работа с книжными и электронными источниками. Выполнения индивидуальных заданий по лабораторному практикуму.	2
ИТОГО часов:				52
Всего:				52

5. ПЕРЕЧЕНЬ УЧЕБНО-МЕТОДИЧЕСКОГО ОБЕСПЕЧЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

5.1. Методические указания для подготовки обучающихся к лекционным занятиям

Лекция является основной формой обучения в высшем учебном заведении. Записи лекций в конспектах должны быть избирательными, полностью следует записывать только определения. В конспекте рекомендуется применять сокращение слов, что ускоряет запись. Вопросы, возникающие в ходе лекции, рекомендуется записывать на полях и после окончания лекции обратиться за разъяснением к преподавателю.

Работа над конспектом лекции осуществляется по этапам:

- повторить изученный материал по конспекту;
- непонятные положения отметить на полях и уточнить;
- неоконченные фразы, пропущенные слова и другие недочеты в записях устранить, пользуясь материалами из учебника и других источников; завершить техническое оформление конспекта (подчеркивания, выделение главного, выделение разделов, подразделов и т.п.).

Самостоятельную работу следует начинать с доработки конспекта, желательно в тот же день, пока время не стерло содержание лекции из памяти. Работа над конспектом не должна заканчиваться с прослушивания лекции. После лекции, в процессе самостоятельной работы, перед тем, как открыть тетрадь с конспектом, полезно мысленно восстановить в памяти содержание лекции, вспомнив ее структуру, основные положения и выводы.

С целью доработки необходимо прочитать записи, восстановить текст в памяти, а также исправить опiski, расшифровать не принятые ранее сокращения, заполнить пропущенные места, понять текст, вникнуть в его смысл. Далее прочитать материал по рекомендуемой литературе, разрешая в ходе чтения, возникшие ранее затруднения, вопросы, а также дополнения и исправляя свои записи. Записи должны быть наглядными, для чего следует применять различные способы выделений. В ходе доработки конспекта углубляются, расширяются и закрепляются знания, а также дополняется, исправляется и совершенствуется конспект. Еще лучше, если вы переработаете конспект, дадите его в новой систематизации записей. Это, несомненно, займет некоторое время, но материал вами будет хорошо проработан, а конспективная запись его приведена в удобный для запоминания вид. Введение заголовков, скобок, обобщающих знаков может значительно повысить качество записи. Этому может служить также подчеркивание отдельных мест конспекта красным карандашом, приведение на полях или на обратной стороне листа краткой схемы конспекта и др.

Подготовленный конспект и рекомендуемая литература используется при подготовке к практическому (лабораторному) занятию. Подготовка сводится к внимательному прочтению учебного материала, к выводу с карандашом в руках всех утверждений и формул, к решению примеров, задач, к ответам на вопросы, предложенные в конце лекции преподавателем или помещенные в рекомендуемой литературе. Примеры, задачи, вопросы по теме являются средством самоконтроля.

Непременным условием глубокого усвоения учебного материала является знание основ, на которых строится изложение материала. Обычно преподаватель напоминает, какой ранее изученный материал и в какой степени требуется подготовить к очередному занятию. Эта рекомендация, как и требование систематической и серьезной работы над всем лекционным курсом, подлежит безусловному выполнению. Потери логической связи как внутри темы, так и между ними приводит к негативным последствиям: материал учебной дисциплины перестает основательно восприниматься, а творческий труд подменяется утомленным переписыванием. Обращение к ранее изученному материалу не только помогает восстановить в памяти известные положения, выводы, но и приводит разрозненные знания

в систему, углубляет и расширяет их. Каждый возврат к старому материалу позволяет найти в нем что-то новое, переосмыслить его с иных позиций, определить для него наиболее подходящее место в уже имеющейся системе знаний. Неоднократное обращение к пройденному материалу является наиболее рациональной формой приобретения и закрепления знаний. Очень полезным, но, к сожалению, еще мало используемым в практике самостоятельной работы, является предварительное ознакомление с учебным материалом. Даже краткое, беглое знакомство с материалом очередной лекции дает многое. Обучающиеся получают общее представление о ее содержании и структуре, о главных и второстепенных вопросах, о терминах и определениях. Все это облегчает работу на лекции и делает ее целеустремленной.

5.2. Методические указания для подготовки обучающихся к лабораторным занятиям

Лабораторные работы сопровождают и поддерживают лекционный курс.

Обучающийся должен подготовиться к выполнению лабораторных работ строго в соответствии с содержанием курса.

В начале каждого лабораторного занятия обучающийся должен ознакомиться с теоретическим материалом, необходимым для выполнения текущей лабораторной работы.

Подготовить ответы на контрольные вопросы, которые соответствуют теме лабораторной работы.

Каждая лабораторная работа содержит список индивидуальных заданий, на выполнение которых обучающийся должен быть готовым.

Полученные результаты выполнения лабораторной работы обучающийся должен уметь оформить и быть готовым к устной защите.

5.3. Методические указания для подготовки обучающихся к практическим занятиям (не предусмотрено)

5.4. Методические указания по самостоятельной работе обучающихся

Самостоятельная работа обучающихся включает в себя выполнение различного рода заданий, которые ориентированы на более глубокое усвоение материала изучаемой дисциплины. По каждой теме дисциплины обучающимся предлагается перечень заданий для самостоятельной работы. К выполнению заданий для самостоятельной работы предъявляются следующие требования: задания должны исполняться самостоятельно и представляться в установленный срок, а также соответствовать установленным требованиям по оформлению. Обучающимся следует:

- руководствоваться графиком самостоятельной работы, определенным на кафедре;
- выполнять все плановые задания, выдаваемые преподавателем для самостоятельного выполнения, и разбирать неясные вопросы на лабораторных и практических занятиях, а также получить информацию на консультациях.

Работа с литературными источниками и интернет ресурсами

В процессе подготовки к практическим занятиям, обучающимся необходимо обратить особое внимание на самостоятельное изучение рекомендованной учебно-методической (а также научной и популярной) литературы.

Самостоятельная работа с учебниками, учебными пособиями, научной, справочной и популярной литературой, материалами периодических изданий и Интернета, статистическими данными является наиболее эффективным методом получения знаний, позволяет

значительно активизировать процесс овладения информацией, способствует более глубокому усвоению изучаемого материала, формирует у обучающихся свое отношение к конкретной проблеме.

Более глубокому раскрытию вопросов способствует знакомство с дополнительной литературой, рекомендованной преподавателем по каждой теме семинарского или практического занятия, что позволяет обучающимся проявить свою индивидуальность в рамках выступления на данных занятиях, выявить широкий спектр мнений по изучаемой проблеме.

Методические указания для подготовки к тестированию

Подготовку к тестированию необходимо осуществлять поэтапно.

На первом этапе необходимо повторить основные положения всех тем, детально разбирая наиболее сложные моменты. Непонятные вопросы необходимо выписывать, чтобы по ним можно было проконсультироваться с преподавателем перед прохождением итогового тестирования. Подготовку по темам каждой дидактической единицы целесообразно производить отдельно. На этом этапе необходимо использовать материалы лекционного курса, материалы семинарских занятий, тестовые задания для текущего контроля знаний, а также презентации лекционного курса.

На втором этапе подготовки предлагается без повторения теоретического материала дать ответы тестовые задания для рубежного контроля знаний. Если ответы на какие-то вопросы вызвали затруднение, необходимо еще раз повторить соответствующий теоретический материал.

Наконец, третий этап подготовки необходимо осуществить непосредственно накануне теста. На данном этапе необходимо аккуратно просмотреть весь лекционный курс.

В случае, если результаты выполнения тестового задания оказались неудовлетворительными, необходимо зафиксировать темы, на вопросы по которым были даны неверные ответы, и еще раз углубленно повторить соответствующие темы в соответствии с указанными выше тремя этапами подготовки к тестированию.

Промежуточная аттестация

По итогам 4 семестра проводится зачет. При подготовке к сдаче зачета рекомендуется пользоваться материалами практических занятий и материалами, изученными в ходе текущей самостоятельной работы.

6. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

№ п/п	Виды учебной работы	Образовательные технологии	Всего часов
1	2	3	4
Семестр 4			
1	1.1 Способы описания алгоритмов	Индивидуальная работа обучающихся по заданиям лабораторного практикума в IDE (IntelliJ IDEA, Eclipse, VS Code). Использование UML-диаграмм (activity diagram, flowchart). Тренинг.	4
2	1.2 Разработка блок-схемы для описания алгоритмов	Самостоятельная работа с использованием онлайн-инструментов для моделирования алгоритмов (draw.io, PlantUML). Построение блок-схемы программы перед реализацией на Java. Тренинг.	4
3	1.3 Разработка псевдокода для описания алгоритмов	Индивидуальная работа по преобразованию алгоритма в псевдокод, а затем в Java-код. Метод мозгового штурма при выборе структур данных и алгоритмов.	4
4	2.1 Использование IDE и инструментов для создания исходных программ	Практическая работа: настройка среды разработки (IntelliJ IDEA/Eclipse). Работа с компилятором javac и JVM. Написание и запуск программ. Тренинг.	4
5	3.1 Работа со стандартной библиотекой Java (Java API)	Практическая работа с коллекциями (ArrayList, HashMap), строками, вводом-выводом (java.io, java.nio). Выполнение заданий по обработке данных. Тренинг.	4
6	4.1 Работа с графическими библиотеками и менеджерами пакетов	Практическая работа с графическими библиотеками (JavaFX, Swing). Подключение внешних библиотек через Maven/Gradle. Разработка простого GUI-приложения.	4
Итого часов в 4 семестре:			24
Всего:			24

7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

7.1 Перечень основной и дополнительной учебной литературы

Список основной литературы

1. ухаметзянов, Р. Р. Основы программирования на Java : учебное пособие / Р. Р. Мухаметзянов. — Набережные Челны : Набережночелнинский государственный педагогический университет, 2017. — 114 с. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/66812.html>). — Режим доступа: для авторизир. пользователей. - DOI: <https://doi.org/10.23682/66812>
2. Вязовик, Н. А. Программирование на Java : учебное пособие / Н. А. Вязовик. — 4-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2025. — 601 с. — ISBN 978-5-4497-0852-6. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/146383.html> . — Режим доступа: для авторизир. пользователей
3. Монахов, В. В. Язык программирования Java и среда NetBeans : учебное пособие / В. В. Монахов. — 4-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2025. — 450 с. — ISBN 978-5-4497-0923-3. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/146413.html> — Режим доступа: для авторизир. пользователей

Список дополнительной литературы

1. Гунько, А. В. Программирование : учебно-методическое пособие / А. В. Гунько. — Новосибирск : Новосибирский государственный технический университет, 2019. — 74 с. — ISBN 978-5-7782-3961-6. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/98810.html> . — Режим доступа: для авторизир. Пользователей
2. Морозова, Ю. В. Практикум по объектно-ориентированному программированию : учебное пособие / Ю. В. Морозова. — Томск : Томский государственный университет систем управления и радиоэлектроники, 2021. — 186 с. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/152837.html> — Режим доступа: для авторизир. пользователей

7.2. Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

<http://elibrary.ru> - Научная электронная библиотека.

7.3. Информационные технологии, лицензионное программное обеспечение

Лицензионное программное обеспечение	Реквизиты лицензий/ договоров
Антивирус Dr.Web Desktop Security Suite	Лицензионный договор № 621 Срок действия: с 25.09.2025 до 24.09.2026
Цифровой образовательный ресурс IPR SMART	Лицензионный договор № 12873/25П от 02.07.2025 г. Срок действия: с 01.07.2025 г. до 30.06.2026 г.
Бесплатное ПО	
Visual Studio Code , LibreOffice, OpenOffice, МойОфис, , Sumatra PDF, 7-Zip, Adobe Acrobat Reader	

8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

8.1. Требования к аудиториям (помещениям, местам) для проведения занятий Учебная аудитория для проведения занятий лекционного типа.

1. Учебная аудитория для проведения занятий лекционного типа

Специализированная мебель: Стол преподавательский - 1шт., компьютерные столы - 10шт., парты - 7шт., стулья - 24шт., доска меловая - 1шт. Лабораторное оборудование, технические средства обучения, служащие для предоставления учебной информации большой аудитории: Персональный компьютер – 10 шт. Экран настенный рулонный – 1 шт.

2. Учебная аудитория для проведения занятий семинарского типа, курсового проектирования (выполнение курсовых работ), групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации

Специализированная мебель: Стол преподавательский - 1шт., доска меловая - 1шт., стул мягкий - 1шт., парты - 6шт., компьютерные столы - 11шт., стулья - 24шт., Лабораторное оборудование, технические средства обучения, служащие для предоставления учебной информации большой аудитории: Персональный компьютер – 7 шт.

3. Лаборатория компьютерной графики

Специализированная мебель: Стол преподавательский - 1шт., компьютерные столы - 10шт., парты - 7шт., стулья - 24шт., доска меловая - 1шт. Лабораторное оборудование, технические средства обучения, служащие для предоставления учебной информации большой аудитории: Персональный компьютер – 10 шт. Экран настенный рулонный – 1 шт.

8.2. Требования к оборудованию рабочих мест преподавателя и обучающихся

1. Рабочее место преподавателя, оснащенное компьютером.
2. Рабочие места обучающихся оснащенные компьютером.

8.3. Требования к специализированному оборудованию - нет

9. ОСОБЕННОСТИ РЕАЛИЗАЦИИ ДИСЦИПЛИНЫ ДЛЯ ИНВАЛИДОВ И ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ

Для обеспечения образования инвалидов и обучающихся с ограниченными возможностями здоровья разрабатывается (в случае необходимости) адаптированная образовательная программа, индивидуальный учебный план с учетом особенностей их психофизического развития и состояния здоровья, в частности применяется индивидуальный подход к освоению дисциплины, индивидуальные задания: рефераты, письменные работы и, наоборот, только устные ответы и диалоги, индивидуальные консультации, использование диктофона и других записывающих средств для воспроизведения лекционного и семинарского материала.

В целях обеспечения обучающихся инвалидов и лиц с ограниченными возможностями здоровья комплектуется фонд основной учебной литературой, адаптированной к ограничению электронных образовательных ресурсов, доступ к которым организован в БИЦ Академии. В библиотеке проводятся индивидуальные консультации для данной категории пользователей, оказывается помощь в регистрации и использовании сетевых и локальных электронных образовательных ресурсов, предоставляются места в читальном зале.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

ПО ДИСЦИПЛИНЕ Системы программирования

1. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ

Системы программирования

Компетенции, формируемые в процессе изучения дисциплины

Индекс	Формулировка компетенции
ПК-3	Способен осуществлять программную реализацию алгоритмов при разработке прикладных информационных систем

2. Этапы формирования компетенции в процессе освоения дисциплины

Основными этапами формирования указанных компетенций при изучении обучающимися дисциплины являются последовательное изучение содержательно связанных между собой разделов (тем) учебных занятий. Изучение каждого раздела (темы) предполагает овладение обучающимися необходимыми компетенциями. Результат аттестации обучающихся на различных этапах формирования компетенций показывает уровень освоения компетенций обучающимися.

Этапность формирования компетенций прямо связана с местом дисциплины в образовательной программе.

Разделы (темы) дисциплины	Формируемые компетенции (коды)
	ПК-3
Раздел 1. Системы программирования для разработки программ на Java.	+
Раздел 2. Компиляторы и интерпретаторы. Интегрированная среда разработки программ	+
Раздел 3. Библиотеки стандартных классов и функций. Инструменты отладки	+
Раздел 4. Графические библиотеки и работа с утилитами	+

3. Показатели, критерии и средства оценивания компетенций, формируемых в процессе изучения дисциплины

ПК- 3 Способен осуществлять программную реализацию алгоритмов при разработке прикладных информационных систем

Индикаторы достижения компетенций	Критерии оценивания результатов обучения				Средства оценивания результатов обучения	
	неудовлетв	удовлетв	хорошо	отлично	Текущий контроль	Промежуточная аттестация
ПК-3.1 Обладает знаниями в области программирования и современных технологий при разработке прикладных информационных систем. Способен осуществлять оптимизацию алгоритмов при создании прикладных программ или прикладных комплексов	Допускает существенные ошибки в знаниях стандартных пакетов прикладных программ для решения практических задач защиты информации, отлаживать и тестировать программы по алгоритмам шифрования; пробелы в знаниях элементов теории чисел.	Демонстрирует некоторые способности в знаниях стандартных пакетов прикладных программ для решения практических задач защиты информации, отлаживать и тестировать программы по алгоритмам шифрования, а также криптографическим методам и средств обеспечения информационной безопасности.	Демонстрирует способность в знаниях стандартных пакетов прикладных программ для решения практических задач защиты информации, отлаживать и тестировать программы по алгоритмам шифрования, а также криптографическим методам и средств обеспечения информационной безопасности	Демонстрирует профессиональную способность к языкам программирования, для целей автоматизации шифрования информации; методов кодирования; знания о государственной политике в области безопасности компьютерных систем; знания элементов теории чисел, составляющие математическую основу криптографических методов.	Текущий тестовый контроль, контрольные вопросы	Зачет
ПК-3.2 Анализирует стандартные технические требования к проектированию программного обеспечения.	Допускает частично освоенное умение пользоваться стандартными программами для решения практических задач защиты информации; отлаживать и тестировать программы по алгоритмам шифрования; применять криптографические методы и средства.	Демонстрирует в целом удовлетворительные, но не систематизированные умения пользоваться стандартными программами для решения практических задач защиты информации; отлаживать и тестировать программы по алгоритмам шифрования; применять криптографические методы	Демонстрирует в целом хорошие, но содержащие отдельные пробелы в использовании стандартных программ для решения практических задач защиты информации; отлаживать и тестировать программы по алгоритмам шифрования; применять элементы теории чисел для моде-	Демонстрирует умения в использовании стандартных программ для решения практических задач защиты информации; отлаживать и тестировать программы по алгоритмам шифрования; применять элементы теории чисел для задач криптографических методов и средств обеспечения информационной безопасности.	Текущий тестовый контроль, контрольные вопросы	Зачет

		чисел для моделирования задач защиты информации.	лирования задач криптографии.			
ПК-3.3 Способен осуществлять разработку технических требований к программному обеспечению, при его создании	Фрагментарно владеет некоторыми пакетами стандартных программ для решения задач защиты информации; не владеет интернет технологиями для организации защиты информации, не владеет возможностями математической основы криптографии теории чисел.	Владеет отдельными навыками работы со стандартными программами для решения задач защиты информации; технологиями отладки и тестирования программ по алгоритмам шифрования; элементов теории чисел для моделирования задач защиты информации.	Демонстрирует в целом успешные знания по стандартным программам для решения практических задач защиты информации; технологии отладки и тестирования программ по алгоритмам шифрования; по криптографическим методам и средствам обеспечения информационной безопасности.	Демонстрирует профессиональные навыки работы в специализированных стандартных пакетах программ для решения практических задач защиты информации; владеет технологиями отладки и тестирования программ по алгоритмам шифрования; владеет способностью применять элементы теории чисел для решения задач криптографии.	Текущий тестовый контроль, контрольные вопросы	Зачет

4. Комплект контрольно-оценочных средств по дисциплине

Контрольные вопросы

по дисциплине Системы программирования

Вопросы к разделу 1.

1. Опишите основные этапы разработки программы на Java — от постановки задачи до получения работающего приложения.
2. Расскажите, как формулируется задача и каким образом разрабатывается её математическая модель.
3. Объясните, как составленный алгоритм преобразуется в программу на языке Java.
4. Опишите различия между процессами отладки и тестирования программ.
5. Расскажите о типах ошибок, которые могут возникать при разработке программ на Java.
6. Объясните назначение юнит-тестирования и особенности его реализации в Java с использованием JUnit.
7. Опишите роль документации и JavaDoc в сопровождении программного продукта.

Вопросы к разделу 2.

1. Расскажите, как происходит процесс компиляции исходного кода на Java в байткод.
2. Опишите принцип работы виртуальной машины Java (JVM), а также особенности интерпретации и JIT-компиляции.
3. Объясните, почему язык Java считается кроссплатформенным.
4. Опишите функции интегрированной среды разработки (IDE) при создании программ на Java.
5. Расскажите о наиболее распространённых средах разработки для языка Java.

Вопросы к разделу 3.

1. Дайте характеристику Java Standard API и объясните его значение для разработчика.
2. Опишите задачи, которые решаются с использованием коллекций и классов пакета `java.util`.
3. Расскажите о возможностях пакета `java.io` для работы с файлами и потоками ввода-вывода.
4. Объясните различия между коллекциями `ArrayList` и `HashMap`.
5. Опишите назначение и применение пакета `java.time` для работы с датами и временем.
6. Расскажите о встроенных средствах отладки, доступных в современных IDE для Java.

Вопросы к разделу 4.

1. Дайте характеристику графических библиотек Java: AWT, Swing и JavaFX.
2. Объясните различия между библиотеками Swing и JavaFX.
3. Расскажите о структуре графического приложения JavaFX, включая понятия сцены и узла.
4. Опишите назначение и функции менеджеров пакетов Maven и Gradle.
5. Объясните, как осуществляется подключение внешних библиотек в проект Java с помощью Maven или Gradle.
6. Расскажите о преимуществах использования внешних библиотек в процессе разработки программного обеспечения.

Вопросы к зачету

по дисциплине «Системы программирования»

1. Понятие системы программирования на примере Java.
2. Стадии разработки программного обеспечения в Java.
3. Технология разработки программ: постановка задачи, алгоритм, кодирование, тестирование, сопровождение.
4. Отличие алгоритма, записанного в псевдокоде, от программы на Java.
5. Уровни языков программирования. Место Java среди них.
6. Преимущества и недостатки низкоуровневых языков по сравнению с Java.
7. Понятие файла в Java. Работа с файловой системой через `java.io` и `java.nio`.
8. Разграничение доступа к файлам средствами операционной системы и Java (права доступа).
9. Характеристика ассемблера и отличие его от высокоуровневых языков (на примере Java).
10. Стандартные классы и функции Java.
11. Правила записи арифметических выражений в Java.
12. Правила записи логических выражений в Java.
13. Назначение методов и подпрограмм в Java.
14. Различие между методами, возвращающими значение (функциями), и процедурами (`void`-методами).
15. Способы задания строковых и символьных констант в Java.
16. Отличие компиляции (`javac`) от интерпретации (JVM).
17. Основные типы данных в Java.
18. Требования к качественным Java-программам (корректность, надёжность, читаемость, модульность).
19. Классификации языков программирования. Парадигмы.
20. Понятие массива в Java. Способы его объявления и использования.
21. Виды массивов в Java (одномерные, многомерные, массивы объектов).
22. Алгоритмы управления вводом-выводом данных в Java.
23. Принципы построения ввода-вывода в Java (`Scanner`, `BufferedReader`, `PrintWriter`).
24. Периферийные устройства и их взаимодействие с программами на Java.
25. Интерфейсы ввода-вывода в Java. Поточковая модель.
26. Управление вводом-выводом средствами Java и ОС.
27. Передача данных с использованием потоков и каналов (`Streams`, `Channels`, `NIO`).
28. Архитектура виртуальной машины Java (JVM).
29. Основные характеристики современных процессоров и их значение для работы JVM.
30. Обзор JIT-компиляции и оптимизаций JVM.
31. Современные процессоры ведущих производителей и их поддержка Java.
32. Нетрадиционные архитектуры (GPU, многопроцессорные системы) и их использование в Java.
33. Нейронные процессоры и возможности их использования через Java-библиотеки.
34. Определения понятий: алфавит, синтаксис, семантика (на примере Java).
35. Особенности языков и систем программирования, применяемых в Java-разработке.
36. Основы объектно-ориентированного программирования (ООП) в Java.

37. Принципы защитного программирования и их реализация в Java (исключения, проверка входных данных).
38. Понятие операционной системы и её функции в контексте запуска Java-программ.
39. Утилиты ОС и их взаимодействие с Java (командная строка, терминал).
40. Доступ к устройствам и файлам средствами Java.
41. Системы счисления, используемые в информатике, и их представление в Java.
42. Способы хранения целых чисел, чисел с плавающей точкой и символов в Java.
43. Организация памяти в JVM: стек, куча (Heap), Metaspace.
44. Механизмы управления памятью в Java.
45. Сборка мусора (Garbage Collection), алгоритмы и принципы работы.
46. Классификация систем памяти в вычислительных системах.
47. Защита памяти в контексте Java и JVM.
48. Алгоритмы замещения информации в кэше и их влияние на производительность Java-программ.
49. Назначение и характеристики вычислительных систем, в которых применяется Java.
50. Современные распределённые вычислительные системы (кластерные и облачные решения) и роль Java.
51. Перспективные направления в архитектурах процессоров и их поддержка в Java (GraalVM, проект Panama).

Список лабораторных работ

по дисциплине «Системы программирования»

Лабораторная 1. Изучение среды программирования. Первая программа на Java

Цель: Освоить создание, компиляцию и запуск программ через IDE и командную строку.

Задания:

1. Вывести ФИО студента.
2. Вывести любимую цитату.
3. Сумма двух чисел.
4. Таблица умножения на число 7.
5. Перевод градусов Цельсия в Фаренгейты.
6. Вывести ASCII-код введённого символа.
7. Первые 10 чисел Фибоначчи.
8. Показать текущие дату и время.
9. Квадрат и куб введённого числа.
10. Нарисовать «лесенку» из звёздочек.

Лабораторная 2. Переменные и типы данных

Цель: Закрепить работу с примитивными типами и строками.

Задания:

1. Создать программу для вычисления суммы целых чисел.
2. Вычислить среднее арифметическое трёх чисел.
3. Преобразовать температуру из Фаренгейтов в Цельсии.
4. Посчитать длину строки, введённой пользователем.
5. Проверить, является ли число положительным, отрицательным или нулём.
6. Найти остаток от деления двух чисел.
7. Обмен значений двух переменных без третьей.
8. Проверить, входит ли число в диапазон [10, 50].
9. Вычислить модуль числа.
10. Сравнить две строки на равенство.

Лабораторная 3. Арифметические и логические операции

Цель: Освоить математические и логические операции в Java.

Задания:

1. Вычислить выражение $(a + b) * c$.
2. Проверить чётность числа.
3. Сравнить два числа и вывести большее.
4. Определить, является ли число делимым на 3 и 5.
5. Проверить, входит ли число в диапазон [1, 100].
6. Вычислить дискриминант квадратного уравнения.
7. Проверить логическое выражение $(x > 0) \ \&\& \ (x < 10)$.
8. Реализовать логическое ИЛИ через условие if.
9. Проверить, является ли символ буквой.
10. Определить знак числа (положительное, отрицательное, ноль).

Лабораторная 4. Условные конструкции (if, switch)

Цель: Освоить ветвление в Java.

Задания:

1. Определить максимальное из трёх чисел.
2. Присвоить оценку по баллам (if-else).
3. Проверить, является ли год високосным.
4. Вывести название дня недели по номеру.
5. Определить, в какой четверти координатной плоскости находится точка.
6. Определить знак числа.
7. Реализовать калькулятор на основе switch.
8. Определить, делится ли число на 2 или 3.
9. Выдать сообщение о возрасте студента.
10. Проверить, входит ли число в заданный диапазон.

Лабораторная 5. Циклы (for, while, do-while)

Цель: Закрепить навыки циклических вычислений.

Задания:

1. Вывести числа от 1 до 100.
2. Найти сумму чисел от 1 до N.
3. Вычислить факториал числа.
4. Вывести таблицу умножения для N.
5. Посчитать количество чётных чисел в диапазоне.
6. Вывести все простые числа до N.
7. Сумма квадратов чисел от 1 до N.
8. Определить, является ли число простым.
9. Вывести треугольник из символов *.
10. Посчитать сумму цифр числа.

Лабораторная 6. Методы и рекурсия

Цель: Научиться создавать функции и использовать рекурсию.

Задания:

1. Метод для нахождения максимума двух чисел.
2. Метод для вычисления факториала рекурсивно.
3. Метод для проверки числа на простоту.
4. Метод для подсчёта суммы элементов массива.
5. Рекурсивная функция для вычисления N-го числа Фибоначчи.
6. Метод для перевода градусов Цельсия в Фаренгейты.
7. Рекурсивная функция для вывода чисел от N до 1.
8. Метод, проверяющий палиндром.
9. Метод для подсчёта количества гласных в строке.
10. Метод для нахождения минимума массива.

Лабораторная 7. Классы и объекты

Цель: Освоить создание классов и объектов.

Задания:

1. Класс «Студент» с полями ФИО, возраст, группа.
2. Класс «Книга» с названием, автором, годом издания.
3. Класс «Прямоугольник» с методами вычисления площади и периметра.
4. Класс «Калькулятор» с методами сложения и вычитания.

5. Класс «Сотрудник» с полями имя, должность, зарплата.
6. Класс «Точка» с координатами x, y.
7. Класс «Круг» с методом вычисления площади.
8. Класс «Машина» с маркой и максимальной скоростью.
9. Класс «Компьютер» с процессором и оперативной памятью.
10. Класс «Часы» с методами отображения времени.

Лабораторная 8. Конструкторы и перегрузка методов

Цель: Научиться создавать конструкторы и перегружать методы.

Задания:

1. Конструкторы для класса «Студент» с разными параметрами.
2. Перегрузка методов сложения чисел.
3. Конструкторы для класса «Книга».
4. Перегрузка метода вычисления площади (квадрат, прямоугольник).
5. Конструкторы класса «Прямоугольник».
6. Перегрузка метода вывода информации о сотруднике.
7. Конструкторы класса «Точка».
8. Перегрузка метода суммирования массивов.
9. Конструкторы класса «Калькулятор».
10. Перегрузка метода отображения времени в 12/24-часовом формате.

Лабораторная 9. Инкапсуляция и модификаторы доступа

Цель: Освоить принципы скрытия данных и контроля доступа.

Задания:

1. Закрытые поля класса «Студент» с геттерами и сеттерами.
2. Инкапсуляция класса «Книга».
3. Приватные поля класса «Прямоугольник».
4. Контроль доступа к зарплате сотрудника.
5. Закрытые поля класса «Машина».
6. Инкапсуляция класса «Калькулятор».
7. Контроль доступа к координатам точки.
8. Приватные методы класса «Часы».
9. Использование модификатора `protected`.
10. Реализация `read-only` свойства для класса «Компьютер».

Лабораторная 10. Наследование

Цель: Освоить наследование и иерархию классов.

Задания:

1. Класс «Студент» → «Аспирант».
2. Класс «Сотрудник» → «Менеджер».
3. Класс «Прямоугольник» → «Квадрат».
4. Класс «Транспорт» → «Машина».
5. Класс «Калькулятор» → «Научный калькулятор».
6. Класс «Животное» → «Собака».
7. Класс «Человек» → «Студент».
8. Класс «Фигура» → «Круг».
9. Класс «Книга» → «Электронная книга».
10. Класс «Компьютер» → «Сервер».

Лабораторная работа №11

Тема: Полиморфизм и абстрактные классы

Цель работы: Освоить работу с абстрактными классами, интерфейсами и полиморфизмом.

Индивидуальные задания:

1. Абстрактный класс Животное → методы `sound()`, `move()`; создать наследников Собака, Кошка.
2. Абстрактный класс Транспорт → метод `start()`, `stop()`; наследники Автомобиль, Велосипед.
3. Интерфейс `Drawable` → методы `draw()`, `resize()`; реализовать в классе Круг.
4. Абстрактный класс Сотрудник → метод `calculateSalary()`; наследники Менеджер, Инженер.
5. Интерфейс `Playable` → метод `play()`; реализовать в МузыкальныйПлеер и ВидеоПлеер.
6. Абстрактный класс Фигура → метод `area()`; наследники Прямоугольник, Квадрат.
7. Абстрактный класс Устройство → метод `turnOn()`, `turnOff()`; наследники Компьютер, Смартфон.
8. Интерфейс `Movable` → метод `move()`; реализовать в Транспорт и Животное.
9. Абстрактный класс Класс → метод `conductLesson()`; наследники Математика, Физика.
10. Интерфейс `Printable` → метод `print()`; реализовать в Документ и Фото.

Лабораторная работа №12

Тема: Работа со строками (`String`, `StringBuilder`)

Цель работы: Освоить методы работы со строками, конкатенацию, поиск, замену, форматирование.

Индивидуальные задания:

1. Подсчитать количество гласных в строке.
2. Проверить, является ли строка палиндромом.
3. Объединить несколько строк в одну с использованием `StringBuilder`.
4. Заменить все пробелы на подчеркивания.
5. Определить длину строки.
6. Вывести все слова строки в обратном порядке.
7. Преобразовать строку в верхний и нижний регистр.
8. Найти позицию первого вхождения символа.
9. Разделить строку на массив слов.
10. Проверить, начинается ли строка с заданного префикса.

Лабораторная работа №13

Тема: Коллекции (`ArrayList`, `HashMap`, `TreeSet`)

Цель работы: Научиться использовать основные коллекции Java для хранения и обработки данных.

Индивидуальные задания:

1. Создать `ArrayList` чисел и найти максимальное значение.

2. Создать `ArrayList` строк и отсортировать по алфавиту.
3. Создать `HashMap` студентов и их оценок; найти среднюю оценку.
4. Создать `TreeSet` чисел и вывести их в отсортированном виде.
5. Удалить повторяющиеся элементы из списка.
6. Подсчитать частоту встречаемости слов в тексте с помощью `HashMap`.
7. Объединить два списка чисел в один `ArrayList`.
8. Проверить наличие ключа в `HashMap`.
9. Найти минимальное и максимальное значение в `TreeSet`.
10. Реализовать очередь с использованием `ArrayList` (`enqueue`, `dequeue`).

Лабораторная работа №14

Тема: Работа с датой и временем (`LocalDate`, `LocalDateTime`)

Цель работы: Освоить работу с датами и временем, вычислять интервалы и форматировать вывод.

Индивидуальные задания:

1. Вывести текущую дату и время.
2. Вычислить разницу в днях между двумя датами.
3. Добавить 10 дней к текущей дате.
4. Вывести день недели для заданной даты.
5. Вывести дату в формате `dd.MM.yyyy`.
6. Определить, является ли год високосным.
7. Вычислить количество месяцев между двумя датами.
8. Получить текущий месяц и год.
9. Преобразовать дату в строку и обратно.
10. Определить, сколько дней осталось до конца года.

Лабораторная работа №15

Тема: Исключения и обработка ошибок

Цель работы: Научиться использовать блоки `try-catch`, создавать собственные исключения и правильно обрабатывать ошибки.

Индивидуальные задания:

1. Деление числа на ноль с обработкой исключения.
2. Чтение числа из строки с обработкой `NumberFormatException`.
3. Открытие файла, которого нет, с обработкой `FileNotFoundException`.
4. Создание собственного исключения `InvalidAgeException`.
5. Проверка ввода пользователя на корректность с использованием исключений.
6. Обработка нескольких исключений в одном блоке `try`.
7. Обработка `NullPointerException`.
8. Создать метод с выбрасыванием собственного исключения при некорректном параметре.
9. Перехват `ArrayIndexOutOfBoundsException` при работе с массивом.
10. Обработка ошибки деления на отрицательное число.

Лабораторная работа №16

Тема: Тестирование программ и документация (JUnit, JavaDoc)

Цель работы: Научиться писать автоматические тесты и создавать документацию для классов и методов.

Индивидуальные задания:

1. Написать JUnit-тест для метода сложения двух чисел.
2. Тест для проверки чётности числа.
3. Тест метода факториала.
4. Тест метода вычисления площади прямоугольника.
5. Тест метода поиска максимума массива.
6. Написать JavaDoc для класса `Студент`.
7. Создать JavaDoc для класса `Калькулятор`.
8. Тест метода перевода градусов Цельсия в Фаренгейты.
9. Создать JavaDoc для класса `Книга`.
10. Написать тесты для метода проверки палиндрома.

Комплект тестовых вопросов

по дисциплине «Системы программирования»

1. Системы для разработки новых программ на Java включают:

- а) системы программирования
- б) программированные системы
- в) системы ретуширования

2. К какому уровню языков относится Java _____

Ответ: Языки высокого уровня

3. Какой язык программирования создан для объектно-ориентированной разработки и получил широкое распространение в 1995 году:

- а) C++
- б) Java
- в) Python

Правильный ответ: б) Java

4. Для чего используют компилятор Java (javac) _____

Ответ: Для перевода исходного кода Java в байткод (.class), который исполняется JVM.

5. Какой язык является машинно-ориентированным:

- а) Java
- б) любой универсальный язык программирования
- в) язык, определяющийся набором команд конкретного процессора

6. Какой язык программирования используется для научных вычислений и был создан раньше Java:

- а) Java
- б) Fortran
- в) Ada

7. Дайте определение система программирования Java _____

Ответ: Совокупность средств разработки на Java: язык программирования, компилятор, интерпретатор JVM, редакторы и отладчики, а также стандартные библиотеки и инструменты тестирования.

8. Выберите верное утверждение о Java:

- а) Программа на Java может работать на любой платформе с установленной JVM
- б) Программы Java работают только на Windows
- в) Java-программы нельзя запускать на сервере

9. Как называются языки, предназначенные для разработки программ, таких как Java:

- а) Языки высокого уровня
- б) Популярные языки
- в) Алгоритмические языки

10. Назовите программы, предназначенные для перевода Java-кода в исполняемый байткод: _____

Ответ: Компиляторы (javac)

11. Из слов какого языка строятся команды Java:

- а) Естественного языка
- б) Логического языка
- в) Алгоритмического языка

12. Одно из самых удобных средств разработки программ на Java:

- а) Ретранслятор
- б) Компилятор и JVM
- в) Интерстеллер

13. Одно из самых удобных средств разработки Java-программ:

- а) Интегрированная среда разработки (IDE)
- б) Интригованная среда разработки
- в) Интегрирующая среда разработки

14. Какой режим работы IDE Java является наиболее удобным для многозадачности:

- а) Тихий режим
- б) Однооконный режим
- в) Многооконный режим

Правильный ответ: в) Многооконный режим

15. Java-программа включает встроенные возможности для:

- а) Ассемблера
- б) Кассемблера
- в) Ассемблятора

16. Одна из популярных IDE для Java:

- а) Turbo Basic
- б) IntelliJ IDEA
- в) Basic S

17. Другая популярная IDE для Java:

- а) Eclipse
- б) Fast Basic
- в) Slow Basic

18. Один из основных языков программирования для веб и серверных приложений является язык программирования _____

Ответ: Java

19. В последнее время получили распространение системы программирования, ориентированные на создание:

- а) Yandex-приложений
- б) Windows-приложений
- в) Веб и серверных приложений на Java

20. Год создания языка Java:

Ответ: 1995

21. Один из программных компонентов системы программирования Java:

- а) Галерея подпрограмм
- б) Редактор текста
- в) Транслятор

22. Компилятор Java предназначен для:

- а) Перевода исходного кода в машинный код конкретного процессора
- б) Перевода исходного кода в байткод JVM
- в) Исполнения кода напрямую

23. Отладчик Java используется для:

- а) Создания интерфейсов
- б) Поиска и исправления ошибок в программе
- в) Компиляции байткода

24. В Java стандартные библиотеки включают:

- а) Галерею подпрограмм
- б) Набор классов java.lang, java.util, java.io
- в) Регулятор соответствующего языка

25. Программа для ввода и редактирования текста в Java IDE:

- а) Компоновщик
- б) Редактор текста
- в) Транслятор

26. Трансляторы Java делятся на _____

Ответ: Компиляторы (javac) и JVM (интерпретатор байткода)

27. Программа, позволяющая управлять процессом исполнения и отлавливать ошибки в Java _____

Ответ: Отладчик (Debugger)

28. В какой аппаратной части ПК хранится информация о конфигурации для запуска JVM _____

Ответ: BIOS / EEPROM

5. Методические материалы, определяющие процедуры оценивания компетенции

5.1 Критерии оценивания качества выполнения лабораторного практикума

Оценка **«зачтено»** выставляется обучающемуся, если лабораторная работа выполнена правильно и обучающийся ответил на все вопросы, поставленные преподавателем на защите.

Оценка **«не зачтено»** выставляется обучающемуся, если лабораторная работа выполнена не правильно или обучающийся не проявил глубоких теоретических знаний при защите работы

5.2 Критерии оценивания качества устного ответа

Оценка **«отлично»** выставляется за глубокое знание предусмотренного программой материала, за умение четко, лаконично и логически последовательно отвечать на поставленные вопросы.

Оценка **«хорошо»** – за твердое знание основного (программного) материала, за грамотные, без существенных неточностей ответы на поставленные вопросы.

Оценка **«удовлетворительно»** – за общее знание только основного материала, за ответы, содержащие неточности или слабо аргументированные, с нарушением последовательности изложения материала.

Оценка **«неудовлетворительно»** – за незнание значительной части программного материала, за существенные ошибки в ответах на вопросы, за неумение ориентироваться в материале, за незнание основных понятий дисциплины.

5.3 Критерии оценивания тестирования

При тестировании все верные ответы берутся за 100%.

90%-100% отлично

75%-90% хорошо

60%-75% удовлетворительно

менее 60% неудовлетворительно

5.4 Критерии оценивания результатов освоения дисциплины

Оценка **«зачтено»** выставляется обучающемуся, если обучающий почти ответил на все вопросы, поставленные преподавателем на защите.

Оценка **«не зачтено»** выставляется обучающемуся, если обучающий не проявил глубоких теоретических знаний при ответе на вопросы.