

УТВЕРЖДЕНА

Решением Ученого совета
АНО ВО «Центральный университет»
«07» марта 2024 г.
Протокол №1

**Рабочая программа дисциплины (модуля)
«Архитектура виртуальных машин»**

Направление подготовки: 02.03.01 Математика и компьютерные науки

Направленность (профиль) подготовки: Разработка

Квалификация (степень) выпускника: бакалавр

Форма обучения: очная

Срок освоения программы: 4 года

Год набора: 2024

**Москва
2024**

Содержание

1. Краткая характеристика дисциплины (модуля)	3
2. Перечень планируемых результатов обучения	4
3. Тематический план	6
4. Содержание дисциплины (модуля)	6
5. Учебно-методическое обеспечение	7
6. Материально-техническое обеспечение	7
7. Методические и оценочные материалы	9

1. Краткая характеристика дисциплины (модуля)

Рабочая программа дисциплины (модуля) «Архитектура виртуальных машин» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования – бакалавриат по специальности 02.03.01 Математика и компьютерные науки, профиль Разработка, утвержденный приказом Министерства науки и высшего образования Российской Федерации № 807 от 23.08.2017 года.

Изучение дисциплины (модуля) «Архитектура виртуальных машин» позволяет понять принципы создания и функционирования виртуализированных сред, что является основой для эффективного использования современных вычислительных ресурсов и облачных технологий. Это знание способствует развитию навыков оптимизации работы систем и повышению безопасности при развертывании программного обеспечения в изолированных средах.

Место дисциплины (модуля) в структуре образовательной программы

Настоящая дисциплина (модуль) включена в учебный план по программе подготовки бакалавриата по направлению 02.03.01 Математика и компьютерные науки, профиль Разработка и входит в вариативную часть Блока 1, формируемую участниками образовательных отношений.

Дисциплина (модуль) является выборной и доступна для изучения на 3 или 4 курсе в 5, 6, 7, 8 семестрах на выбор. Прохождение возможно при условии успешного завершения дисциплин: «Архитектура компьютера и операционные системы», «Дизайн компиляторов».

Цель изучения дисциплины (модуля): формирование знаний и навыков проектирования, реализации и эффективного использования виртуализированных вычислительных сред для оптимизации работы и управления ресурсами современных информационных систем.

Задачи изучения дисциплины (модуля) направлены на формирование у студентов следующий знаний, умений и навыков:

- знание принципов построения архитектуры команд (ISA) виртуальных машин;
- знание устройства выполнения кода на виртуальной машине Java;
- знание алгоритмов автоматического управления динамической памятью (Garbage Collection);
- знание принципов JIT-компиляции;
- умение реализовать выполнение простого набора команд байткода;
- умение реализовать алгоритм уборки мусора;
- умение реализовать генерацию машинного кода;
- навык реализации среды выполнения для простого языка программирования.

2. Перечень планируемых результатов обучения

Компетенции, формируемые в результате освоения дисциплины (модуля) при проведении учебных занятий в форме контактной работы обучающихся с педагогическими работниками Университета и в форме самостоятельной работы обучающихся:

Компетенция	Содержание компетенции	Индикатор компетенции	Перечень планируемых результатов обучения по дисциплине (модулю)
УК-1.	Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.1.	Знает методы поиска и анализа информации в области разработки, основные принципы критической оценки источников информации и их релевантности
		УК-1.2.	Умеет критически оценивать источники информации и синтезировать данные из различных источников для решения задач, применять системный подход к анализу и решению комплексных проблем
		УК-1.3.	Имеет практический опыт работы с современными инструментами и технологиями для обработки информации, формулировании и структурировании задач на основе полученной информации
ОПК-4.	Способен находить, анализировать, реализовывать программно и использовать на практике математические алгоритмы, в том числе с применением современных вычислительных систем	ОПК-4.1.	Знает базовые основы современного математического аппарата, связанного с проектированием, разработкой, реализацией и оценкой качества программных продуктов и программных комплексов в различных областях человеческой деятельности
		ОПК-4.2.	Умеет использовать этот математический аппарат в профессиональной деятельности
		ОПК-4.3.	Имеет практический опыт применения современного математического аппарата, связанного с проектированием, разработкой, реализацией и оценкой качества программных продуктов и программных комплексов в различных областях человеческой деятельности
ОПК-5.	Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной	ОПК-5.1.	Знает технологии, необходимые для прикладного программирования, включая современные функциональные языки программирования, а

	деятельности		также основные принципы и понятия, применяемыми при использовании компьютерных сетей
		ОПК-5.2.	Умеет пользоваться технологиями прикладного программирования, включая среды высокоуровневого программирования
		ОПК-5.3.	Имеет практический опыт использования технологий прикладного программирования
ПК-3.	Способен применять методы математического и алгоритмического моделирования для решения как теоретических, так и практических задач в рамках профессиональной деятельности	ПК-3.1.	Знает основные методы математического и алгоритмического моделирования, а также их применение для решения теоретических и прикладных задач
		ПК-3.2.	Умеет применять методы математического и алгоритмического моделирования для анализа и решения различных задач в области математики и компьютерных наук
		ПК-3.3.	Имеет опыт использования методов математического и алгоритмического моделирования при решении теоретических и прикладных задач в профессиональной деятельности

3. Тематический план

№п/п	Наименование раздела дисциплины (модуля)	Трудоемкость, академические часы				ТКУ (текущий контроль успеваемости)
		<i>Очная форма</i>				
		Контактная работа		Контроль	Самостоятельная работа	
Лекции	Семинары (практические занятия)					
1	Архитектура набора команд байткода и его выполнение	8	8		43	Домашние задания, Контрольная работа
2	Автоматическое управление памятью	8	8		44	Домашние задания, Контрольная работа
3	JIT-компиляция	8	8		43	Домашние задания, Контрольная работа
	<i>Зачет с оценкой</i>			12		Проект
	Итого:	24	24	12	130	
	<i>Объем дисциплины (модуля) (в ак. ч.)</i>	190				
	<i>Объем дисциплины (модуля) (в зач. ед.)</i>	5				

4. Содержание дисциплины (модуля)

№п/п	Наименование раздела дисциплины (модуля)	Содержание дисциплины (модуля) по темам
1	Архитектура набора команд байткода и его выполнение	Платформонезависимый байткод. Оптимизированный набор команд. Команды для памяти, стеков и вызовов. Способы выполнения байткода виртуальной машиной. Переносимость приложений.
2	Автоматическое управление памятью	Автоматическое выделение и освобождение. Сборщик мусора. Основные алгоритмы: подсчет ссылок, маркировка, копирование. Уменьшение ошибок памяти. Влияние на производительность.
3	JIT-компиляция	Преобразование байткода в машинный код. Оптимизация горячих участков. Повышение скорости исполнения. Сочетание переносимости и эффективности. Адаптивная оптимизация.

5. Учебно-методическое обеспечение

Университет располагает полным набором лицензионного и свободно распространяемого программного обеспечения, включая продукты отечественного производства.

Каждый студент в течение всего периода обучения получает индивидуальный неограниченный доступ к электронно-библиотечной системе и электронной информационно-образовательной среде университета. Эти системы предоставляют возможность доступа к ресурсам из любой точки, где есть подключение к сети Интернет, как на территории университета, так и за его пределами.

Студентам обеспечен удаленный доступ к современным профессиональным базам данных и информационным справочным системам.

Основная литература:

1. Малявко А. А. Формальные языки и компиляторы : учебник для вузов / А. А. Малявко. — Москва : Издательство Юрайт, 2025. — 429 с. — (Высшее образование). — ISBN 978-5-534-04288-7. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/562822>.

Дополнительная литература:

1. Новожилов О. П. Архитектура ЭВМ и вычислительных систем : учебник для вузов / О. П. Новожилов. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2025. — 505 с. — (Высшее образование). — ISBN 978-5-534-20365-3. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/568920>.

2. Толстобров А. П. Архитектура ЭВМ : учебник для вузов / А. П. Толстобров. — 3-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 162 с. — (Высшее образование). — ISBN 978-5-534-16839-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/566711>.

6. Материально-техническое обеспечение

Университет располагает материально-технической базой, соответствующей действующим противопожарным правилам и нормам и обеспечивающей проведение всех видов дисциплинарной и междисциплинарной подготовки, практической и научно-исследовательской работ обучающихся, предусмотренных учебным планом.

Помещения, которые представляют собой учебные аудитории для проведения занятий лекционного типа, занятий семинарского (практического) типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, а также помещения для самостоятельной работы и помещения для хранения и профилактического обслуживания учебного оборудования. Помещения укомплектованы специализированной мебелью и техническими средствами обучения, служащими для представления учебной информации большой аудитории.

Изучение дисциплины (модуля) обеспечивается в учебных аудиториях, оснащенных:

- столами и стульями;
- компьютерной техникой;
- специализированным оборудованием, включая демонстрационное оборудование.

Помещения для самостоятельной работы обучающихся, в том числе приспособленные для использования инвалидами и лицами с ограниченными возможностями здоровья, оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа к в электронную информационно-образовательную среду Университета.

Обучающимся предоставляется доступ (в том числе удаленный) к ресурсам информационно-телекоммуникационной сети «Интернет», электронным ресурсам (в том

числе электронным библиотечным системам, современным профессиональным базам данных и информационным справочным системам):

№	Наименование портала (издания, курса, документа)	Ссылка
1.	Научная электронная библиотека eLibrary.ru библиотека	https://elibrary.ru/defaultx.asp
2.	База данных для IT-специалистов	https://habr.com
3.	База данных ScienceDirect	https://www.sciencedirect.com
4.	Официальный сайт Министерства науки и высшего образования Российской Федерации	https://minobrnauki.gov.ru/
5.	Федеральный портал «Российское образование»	https://www.edu.ru/
6.	Информационная система "Единое окно доступа к образовательным ресурсам"	http://window.edu.ru/
7.	Единая коллекция цифровых образовательных ресурсов	http://school-collection.edu.ru/
8.	Федеральный центр информационно - образовательных ресурсов	http://fcior.edu.ru/

Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), в том числе комплект лицензионного программного обеспечения, современные профессиональные базы данных и информационные справочные системы:

Наименование ПО	Производство	Лицензионное / свободно распространяемое
Операционные системы:		
Microsoft Imagine (Windows Client, Server)	зарубежное	лицензионное
Браузеры:		
Яндекс.Браузер	отечественное	свободно распространяемое
Google Chrome	зарубежное	свободно распространяемое
Офисные приложения:		
Microsoft Imagine (Visio, OneNote)	зарубежное	лицензионное
TeXstudio	зарубежное	свободно распространяемое
Adobe Acrobat Reader	зарубежное	свободно распространяемое
Программное обеспечение для планирования и учета времени:		
Toggle app	зарубежное	свободно распространяемое
Системы управления проектами:		
Microsoft Imagine (Project)	зарубежное	лицензионное
Системы управления базами данных:		
Microsoft Imagine (SQL Server)	зарубежное	лицензионное
Системы резервного копирования (backup):		
Acronis Backup Advanced for HyperV	зарубежное	лицензионное
Справочно-правовые системы:		
КонсультантПлюс: справочно-правовая система	отечественное	лицензионное
Средства антивирусной защиты:		
Kaspersky Endpoint Security для бизнеса Стандартный Russian Edition	отечественное	лицензионное
Среды разработки:		
Visual Studio Code	зарубежное	свободно распространяемое
Bash (Unix shell)	зарубежное	свободно распространяемое
Anaconda	зарубежное	свободно распространяемое
Robotic Operating System	зарубежное	свободно распространяемое

CopelliaSim	зарубежное	свободно распространяемое
Google Colaboratory	зарубежное	свободно распространяемое
Пакеты программных средств и библиотек:		
AutoPsy	зарубежное	свободно распространяемое
Interactive Disassembler (IDA)	зарубежное	свободно распространяемое
Системы управления библиографической информацией:		
Zotero	зарубежное	свободно распространяемое
Сервисы и службы:		
Bind	зарубежное	свободно распространяемое
Docker	зарубежное	свободно распространяемое

7. Методические и оценочные материалы

Методические указания для обучающихся по освоению дисциплины (модуля)

В процессе изучения дисциплины (модуля) «Архитектура виртуальных машин» в рамках текущего контроля успеваемости используются такие виды учебной работы, как лекции, семинары, контрольные работы, домашние задания, проект, а также различные виды самостоятельной работы обучающихся по заданию преподавателя, направленные на развитие навыков профессиональной лексики, закрепление практических профессиональных компетенций, поощрение инициатив.

Лекция – систематическое, последовательное, монологическое изложение преподавателем учебного материала, как правило, теоретического характера.

В процессе лекций рекомендуется вести конспект лекций: кратко и схематично фиксировать основные идеи, выводы и обобщения лекции; выделять важные мысли, ключевые слова и термины. Необходимо отметить вопросы или материалы, которые вызывают затруднения, и попытаться найти ответы в рекомендованной литературе. Если разобраться в материале не удастся, следует сформулировать вопрос и задать его преподавателю на консультации или во время семинарского (практического) занятия.

Участие в семинаре – активная работа студента на семинаре, его ответы на вопросы преподавателя и участие в дискуссии.

Для успешного участия в семинаре студентам рекомендуется заранее ознакомиться с темой обсуждения, прочитать необходимые материалы и подготовить вопросы. Важно активно слушать и вовлекаться в дискуссию, высказывая свои мнения и аргументируя их. При ответах на вопросы преподавателя стоит быть уверенным, четким и логичным, опираясь на изученный материал. Также полезно поддерживать диалог с однокурсниками, чтобы обогатить обсуждение и расширить свои знания.

Домашнее задание – набор задач по темам недели.

При работе над домашними заданиями важно внимательно ознакомиться с требованиями и сроками выполнения. Рекомендуется разбивать задания на этапы, чтобы избежать перегрузки и лучше усвоить материал. Использовать различные источники информации, включая учебники и онлайн-ресурсы, для более глубокого понимания темы.

Контрольная работа – письменная работа с набором задач, которые нужно решить за ограниченное время.

Цель контрольной работы - получить специальные знания по одной или нескольким темам дисциплины (модуля) и продемонстрировать навыки их практического применения.

Проект – исследовательская работа по курсу и презентация результатов.

Для успешной подготовки к проекту: четко определите цели и задачи проекта, распределите роли и обязанности между участниками, а также установите сроки

выполнения каждой части работы. Регулярно проводите встречи для обсуждения прогресса и решения возникающих вопросов.

Самостоятельная работа – работа студентов, направленная на углубленное изучение отдельных тем и вопросов учебной дисциплины (модуля).

В процессе самостоятельной работы студенты взаимодействуют с рекомендованными материалами при минимальном участии преподавателя. Задачи студента включают работу с конспектами лекций (обработка текста), повторное изучение учебных материалов, планов и тезисов ответов, изучение дополнительных тем, выполнение учебно-исследовательских заданий и другое.

Система оценивания результатов обучения по дисциплине (модулю)

Критерии получения уровня и оценивания сформированности компетенций по дисциплине (модулю) «Архитектура виртуальных машин»

Оценивание уровня учебных достижений, обучающихся по дисциплине (модулю), осуществляется в виде текущего контроля успеваемости и промежуточной аттестации.

Промежуточная аттестация по дисциплине (модулю) осуществляется в форме *зачета с оценкой*, при этом проводится оценка компетенций, сформированных по дисциплине.

Для оценивания текущего контроля успеваемости и промежуточной аттестации используется десятибалльная шкала оценивания, которая соотносится с традиционной пятибалльной шкалой следующим образом:

Десятибалльная оценка	Пятибалльная оценка	Оценка за зачет	Общая характеристика результата обучения по дисциплине (модулю)
10	Отлично	Зачтено	Студент полностью владеет знаниями, изложенными в рабочей программе, и глубоко осмысляет дисциплину. Он самостоятельно и логически последовательно отвечает на все вопросы, акцентируя внимание на наиболее важном. Умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал, выделяя ключевые моменты и устанавливая причинно-следственные связи. Четко формулирует ответы, уверенно интерпретирует результаты анализов и других исследований, а также решает сложные задачи. Студент хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет связывать теоретические аспекты дисциплины (модуля) с практическими задачами.
9	Отлично	Зачтено	
8	Отлично	Зачтено	
7	Хорошо	Зачтено	Студент обладает знаниями предмета почти в полном объеме рабочей программы и самостоятельно, логически
6	Хорошо	Зачтено	

Десятибалльная оценка	Пятибалльная оценка	Оценка за зачет	Общая характеристика результата обучения по дисциплине (модулю)
			последовательно и всесторонне отвечает на все вопросы, акцентируя внимание на наиболее значимых моментах. Он умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал, выделяя его ключевые аспекты и устанавливая причинно-следственные связи. Формулирует свои ответы, уверенно интерпретирует результаты анализов и других исследований, а также решает сложные ситуационные задачи. Студент хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет связывать теоретические аспекты предмета с практическими задачами.
5	Удовлетворительно	Зачтено	Студент обладает базовыми знаниями по дисциплине (модулю), но испытывает трудности при самостоятельных ответах и использует неточные формулировки. В ходе ответов он допускает ошибки, касающиеся сути вопросов. Студент способен решать только самые простые задачи и владеет лишь минимальным набором методов исследования.
4	Удовлетворительно	Зачтено	
3	Не сдан	Не зачтено	Студент не овладел обязательным минимумом знаний по предмету и не может ответить на вопросы, даже если преподаватель задает дополнительные наводящие вопросы.
2	Не сдан	Не зачтено	
1	Не сдан	Не зачтено	

Дисциплина (модуль) «Архитектура виртуальных машин» оценивается следующим образом:

Активность	Вес	Количество	Описание
Домашние задания	35%	14	Набор задач по темам недели
Контрольные работы	25%	3	Письменная работа с набором задач, которые нужно решить за ограниченное время
Зачет с оценкой	40%	1	Защита итогового проекта

Формула расчёта итоговой оценки по дисциплине (модулю) «Архитектура виртуальных машин»: « $0,35 \times$ среднее за домашние задания + $0,25 \times$ среднее за контрольные работы + $0,4 \times$ зачет с оценкой».

Текущий контроль успеваемости обучающихся по дисциплине (модулю)

Примерные домашние задания

Домашнее задание по теме «Платформонезависимый байткод»

1. Исследуйте и опишите структуру байткода Java или .NET IL, укажите, как обеспечивается его платформонезависимость.
2. Сравните байткод и машинный код с точки зрения переносимости и эффективности.
3. Найдите примеры команд байткода, которые обеспечивают независимость от архитектуры процессора.
4. Проанализируйте, как байткод обрабатывает различия в размерах типов данных на разных платформах.
5. Напишите краткий доклад о роли байткода в обеспечении кроссплатформенности современных языков программирования.

Домашнее задание по теме «Способы выполнения байткода виртуальной машиной»

1. Опишите и сравните основные методы выполнения байткода: интерпретация, JIT-компиляция и AOT-компиляция.
2. Проведите эксперимент: запустите простой байткод на виртуальной машине с включенным и выключенным JIT, сравните время выполнения.
3. Нарисуйте схему работы виртуальной машины при интерпретации байткода.
4. Исследуйте механизм стека вызовов в виртуальной машине при выполнении байткода.
5. Подготовьте презентацию о том, как виртуальная машина управляет памятью во время выполнения байткода.

Домашнее задание по теме «Основные алгоритмы: подсчёт ссылок, маркировка, копирование»

1. Реализуйте на псевдокоде алгоритм подсчёта ссылок и опишите его основные этапы.
2. Сравните алгоритмы маркировки и копирования по эффективности и применимости.
3. Проанализируйте, как циклические ссылки влияют на работу алгоритма подсчёта ссылок.
4. Нарисуйте схему работы алгоритма маркировки и удаления мусора.
5. Напишите эссе о том, как выбор алгоритма сборки мусора влияет на производительность и стабильность приложения.

Домашнее задание по теме «Преобразование байткода в машинный код»

1. Опишите этапы преобразования байткода в машинный код при JIT-компиляции.
2. Исследуйте, какие оптимизации могут быть применены во время преобразования байткода в машинный код.
3. Напишите пример простого байткода и покажите, как он может быть преобразован в машинный код.
4. Проанализируйте преимущества и недостатки преобразования байткода в машинный код в рантайме.
5. Подготовьте доклад о том, как адаптивная оптимизация улучшает качество сгенерированного машинного кода.

Примерные задания по контрольным работам

Контрольная работа № 1

Задание 1. Объясните, что такое платформонезависимый байткод и почему он важен для переносимости приложений.

Задание 2. Назовите и опишите основные типы команд байткода, используемые для работы с памятью.

Задание 3. Какие команды байткода применяются для управления стеком и вызовами функций? Приведите примеры.

Задание 4. Опишите, как виртуальная машина выполняет байткод: основные этапы и механизмы.

Задание 5. В чем заключается оптимизация набора команд байткода и как она влияет на производительность?

Задание 6. Сравните выполнение байткода на виртуальной машине с интерпретацией и компиляцией.

Задание 7. Приведите пример команды байткода и объясните, как она изменяет состояние виртуальной машины.

Задание 8. Какие проблемы переносимости решает использование байткода?

Задание 9. Опишите, как стек и память взаимодействуют при выполнении команд байткода.

Задание 10. Объясните, почему набор команд байткода должен быть оптимизированным и каким образом это достигается.

Контрольная работа № 2

Задание 1. Что такое сборщик мусора и какую роль он играет в управлении памятью?

Задание 2. Опишите алгоритм подсчёта ссылок и его преимущества и недостатки.

Задание 3. Как работает алгоритм маркировки и удаления объектов?

Задание 4. Объясните принцип работы алгоритма копирования в сборщиках мусора.

Задание 5. Какие ошибки памяти могут быть уменьшены с помощью автоматического управления памятью?

Задание 6. Как сборщик мусора влияет на производительность приложений?

Задание 7. В чем разница между ручным и автоматическим управлением памятью?

Задание 8. Объясните, что такое циклические ссылки и как они влияют на подсчёт ссылок.

Задание 9. Назовите основные этапы работы сборщика мусора.

Задание 10. Какие методы оптимизации применяются для уменьшения влияния сборщика мусора на производительность?

Контрольная работа № 3

Задание 1. Что такое JIT-компиляция и как она отличается от традиционной компиляции?

Задание 2. Объясните процесс преобразования байткода в машинный код при JIT-компиляции.

Задание 3. Что такое "горячие участки" кода и почему их оптимизация важна?

Задание 4. Как JIT-компиляция сочетает переносимость и эффективность исполнения?

Задание 5. Опишите, что такое адаптивная оптимизация в контексте JIT.

Задание 6. Какие преимущества даёт использование JIT-компиляции для производительности приложений?

Задание 7. Приведите пример, когда JIT-компиляция может ухудшить производительность.

Задание 8. Как JIT-компиляция взаимодействует с виртуальной машиной?

Задание 9. Какие типы оптимизаций обычно применяются при JIT-компиляции?

Задание 10. Объясните, почему JIT-компиляция считается компромиссом между интерпретацией и статической компиляцией.

Примерное описание и критерии оценивания к итоговому проекту

Описание проекта:

В рамках итогового проекта студенты должны разработать прототип виртуальной

машины (VM), способной выполнять платформонезависимый байткод с использованием оптимизированного набора команд, реализовать автоматическое управление памятью и внедрить базовую JIT-компиляцию для повышения производительности.

Основные задачи проекта:

1. Архитектура байткода и его выполнение

- Спроектировать компактный и оптимизированный набор команд байткода, включающий операции работы с памятью, стеком и вызовами.

- Обеспечить платформонезависимость байткода и переносимость приложений.

- Реализовать интерпретатор, выполняющий байткод.

2. Автоматическое управление памятью

- Реализовать механизм автоматического выделения и освобождения памяти.

- Внедрить один из основных алгоритмов сборки мусора (подсчёт ссылок, маркировка или копирование).

- Продемонстрировать влияние выбранного алгоритма на производительность и устойчивость работы VM.

3. JIT-компиляция

- Реализовать простой JIT-компилятор, преобразующий байткод в машинный код.

- Оптимизировать горячие участки кода для повышения скорости исполнения.

- Обеспечить сочетание переносимости (через байткод) и эффективности (через JIT).

- Подготовить анализ адаптивных оптимизаций и их влияния на производительность.

4. Документация и демонстрация

- Подготовить техническое описание архитектуры VM, используемых алгоритмов и решений.

- Провести тестирование и предоставить сравнительный анализ производительности с и без JIT, а также с разными методами управления памятью.

- Представить проект в виде презентации с демонстрацией работы.

Критерии оценивания:

- Архитектура байткода
- Реализация интерпретатора
- Автоматическое управление памятью
- JIT-компиляция
- Документация и презентация
- Тестирование и анализ результатов

Задания для промежуточной аттестации по дисциплине (модулю)

№ п/п	Задание	Ответ	Компетенция
1.	Что такое байткод?	Платформонезависимый промежуточный код	ОПК-5
2.	Какая из команд байткода обычно отвечает за работу со стеком? a) LOAD b) PUSH c) CALL d) JMP	b	ОПК-5
3.	Какой из методов сборки мусора основан на подсчёте ссылок? a) Маркировка и очистка b) Копирование c) Подсчёт ссылок d) Генерационный сборщик	c	ОПК-4

4.	<p>Что из перечисленного является преимуществом JIT-компиляции?</p> <p>а) Полная независимость от платформы</p> <p>б) Исключение необходимости сборщика мусора</p> <p>с) Уменьшение времени компиляции исходного кода</p> <p>д) Повышение скорости исполнения за счёт преобразования байткода в машинный код во время выполнения</p>	d	ПК-3
5.	<p>Укажите наиболее распространённый способ выполнения байткода в современных виртуальных машинах.</p>	JIT-компиляция	ПК-3
6.	<p>Какое влияние сборщик мусора оказывает на производительность программы?</p> <p>а) Всегда ускоряет выполнение</p> <p>б) Может вызывать паузы и замедлять выполнение</p> <p>с) Не влияет на производительность</p> <p>д) Уменьшает потребление памяти без затрат времени</p>	b	ОПК-5
7.	<p>Что обеспечивает переносимость приложений, использующих байткод?</p> <p>а) Наличие виртуальной машины, реализующей выполнение байткода на разных платформах</p> <p>б) Использование одного и того же исходного кода на всех платформах</p> <p>с) Компиляция в машинный код каждой платформы заранее</p> <p>д) Использование только стандартных библиотек</p>	a	УК-1
8.	<p>Укажите алгоритм сборки мусора, который предполагает разделение памяти на «активную» и «пассивную» области.</p>	Копирование	ОПК-4
9.	<p>Назовите основную структуру данных, используемую для выполнения команд байткода.</p>	Стек	УК-1
10.	<p>Как называется процесс автоматического освобождения неиспользуемой памяти?</p>	Сборка мусора	ОПК-5
11.	<p>Какой термин описывает процесс преобразования байткода в машинный код во время выполнения программы?</p>	JIT-компиляция	ПК-3
12.	<p>Назовите один из основных алгоритмов сборщика мусора, основанный на обходе объектов и отметке достижимости.</p>	Маркировка и очистка	ОПК-4