

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ПОВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ



УТВЕРЖДАЮ
Декан ФИиВТ

УТВЕРЖДАЮ /А.А. Кречетов/
(Ф.И.О. декана (директора института))

14.02.2024 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Б.1.2.6 Системное программное обеспечение

(код и наименование дисциплины по учебному плану)

Направление подготовки
(специальность)

09.03.01 Информатика и вычислительная техника

Квалификация выпускника

Бакалавр

(бакалавр/магистр/специалист)

Направленность

Вычислительные машины, комплексы, системы и сети

Курс 3
Семестр 6

Распределение учебного времени

Трудоемкость по учебному плану	108 / 3	часов/зачетных единиц
Лекции	18	часов
Лабораторные работы	36	часов
Практические занятия	-	часов
Иная контактная работа	-	часов
Всего контактной работы (без учета экз.)	54	часов
Контактная работа по экзамену	-	часов
Курсовой проект (работа)	-	семестр
Самостоятельная работа обучающихся (без учета экз.)	54	часов
Самостоятельная работа по подготовке к экзамену	-	часов
Экзамен	-	семестр
Зачет	6	семестр
БРК, ДЗ	-	семестр

(год)

Программа составлена в соответствии с требованиями ФГОС ВО направления подготовки (специальности) 09.03.01 Информатика и вычислительная техника

Программу составили:

заведующий кафедрой с ученой степенью кандидата наук	ИВС	СОГЛАСОВАНО	Д.В. Морохин
(должность)	(кафедра)		(И.О. Фамилия)

РАССМОТРЕНА и ОДОБРЕНА на заседании кафедры, за которой закреплена дисциплина
Кафедра информационно-вычислительных систем

(наименование кафедры)		
06.02.2024	протокол №	20
(дата)		

Заведующий кафедрой	СОГЛАСОВАНО	Д.В. Морохин
		(И.О. Фамилия)

Рабочая программа СОГЛАСОВАНА с факультетом (институтом), выпускающей(ими)
кафедрой(ами).

СООТВЕТСТВУЕТ действующей ОП.

Заведующий кафедрой	СОГЛАСОВАНО	Д.В. Морохин
		(И.О. Фамилия)

Председатель методической комиссии факультета (института), в который входит
выпускающая кафедра

СОГЛАСОВАНО	А.А. Кречетов
	(И.О. Фамилия)

Эксперт(ы): Чернов Михаил Павлович, Заместитель генерального директора по
производству ЗАО СКБ "Хроматэк"

Рабочая программа проверена и зарегистрирована в УМЦ 21.02.2024 г.

Специалист учебно-методического центра СОГЛАСОВАНО /Т.А. Смирнова/

Раздел 1. ЦЕЛЬ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Целью освоения дисциплины является достижение планируемых результатов обучения, соответствующих установленным в ОПОП индикаторам достижения компетенций:

Код и наименование компетенции	Код и наименование индикатора достижения компетенции	Результаты обучения
1. ПК-3 Способен разрабатывать компоненты системных программных продуктов	ПК-3.1. Знать: основы архитектуры, устройства и функционирования <u>вычислительных систем</u>	знания: Знать: основы архитектуры, устройства и функционирования вычислительных систем умения: навыки:
	ПК-3.2. Уметь: применять языки программирования, определенные в техническом задании для написания программного кода	знания: умения: Уметь: применять языки программирования, определенные в техническом задании для написания программного кода навыки:
	ПК-3.3. Владеть навыками: разработки исходного кода и создания бинарных файлов программного обеспечения создаваемых программных средств	знания: умения: навыки: Владеть навыками: разработки исходного кода и создания бинарных файлов программного обеспечения создаваемых программных средств

Раздел 2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина относится к части, формируемой участниками образовательных отношений ОПОП.

Дисциплина является обязательной

Изучаемая дисциплина является основой для продолжения формирования указанных компетенций в следующих дисциплинах: Микропроцессорные системы (ПК-3); практиках: Производственная практика. Технологическая (проектно-технологическая) практика (ПК-3); государственной итоговой аттестации в форме: Подготовка к процедуре защиты и защита выпускной квалификационной работы (ПК-3)

Раздел 3. ОПИСАНИЕ ОБРАЗОВАТЕЛЬНЫХ ТЕХНОЛОГИЙ

Для формирования заявленных компетенций используются методологические технологии, реализующие деятельностный, личностно-ориентированный, практико-ориентированный подходы.

Основными стратегическими технологиями являются: лекционные занятия, практические и лабораторные занятия, процедуры самообучения

На достижение конкретных целей обучения направлены применяемые тактические технологии: задания, информационные, классическая лекция

Раздел 4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

6 семестр

Виды и темы занятий	Количество часов	Формируемые компетенции
Состав, структура и основные функции СПО.	8	ПК-3
Лекция. Иерархическая структура СПО.	1	
Лекция. Операционные системы. Файловые системы.	1	
Задания для самостоятельной работы, в том числе выполнение Проработка лекций. Выполнение тестов для самоконтроля.	6	
Основные принципы построения трансляторов	17	ПК-3
Лекция. Этапы трансляции. Общая схема работы транслятора.	1	
Лекция. Лексические анализаторы.	1	
Лекция. Синтаксические анализаторы. Синтаксически управляемый перевод.	2	
Лабораторная работа. Работа с таблицами символов.	4	
Задания для самостоятельной работы, в том числе выполнение Проработка лекций. Подготовка к лабораторным работам. Выполнение тестов для самоконтроля.	9	
Языки и цепочки символов	18	ПК-3
Лекция. Методы задания языков	2	
Лекция. Особенности языков программирования	2	
Лабораторная работа. Основных понятия теории регулярных грамматик. Построение лексического анализатора	2	
Задания для самостоятельной работы, в том числе выполнение Проработка лекций. Выполнение тестов для самоконтроля. Подготовка к лабораторным работам.	12	
Теория грамматик	37	ПК-3
Лекция. Формальное определение грамматики. Классификация языков и грамматик. Однозначные грамматики.	2	
Лекция. Процесс вывода. Распознаватели.	2	
Лабораторная работа. Построение синтаксического анализатора	18	
Задания для самостоятельной работы, в том числе выполнение Проработка лекций. Выполнение тестов для самоконтроля. Подготовка к лабораторным работам.	15	
Генерация и оптимизация кода	14	ПК-3
Лекция. Семантический анализ. Синтаксически управляемый перевод. Способы внутреннего представления программ.	1	
Лекция. Оптимизация кода.	1	
Лабораторная работа. Генерация и оптимизация объектного кода.	6	
Задания для самостоятельной работы, в том числе выполнение Проработка лекций. Выполнение тестов для самоконтроля. Подготовка к лабораторным работам.	6	
Методы тестирования	14	ПК-3

Лекция. Методы тестирования. Основные понятия.	1
Лекция. Подходы к тестированию. Организация процесса тестирования ПО.	1
Лабораторная работа. Тестирование ПО по методу «белого ящика» задач вычислительного типа.	3
Лабораторная работа. Тестирование ПО по методу «черного ящика» задач по обработке информации	3
Задания для самостоятельной работы, в том числе выполнение Проработка лекций. Выполнение тестов для самоконтроля. Подготовка к лабораторным работам.	6
Иная контактная работа:	0

Раздел 5. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Изучение дисциплины рекомендуется начать с ознакомления с рабочей программой, ее структурой и содержанием разделов. Учебный материал структурирован, изучение дисциплины осуществляется в тематической последовательности. **Занятия лекционного типа** дают систематизированные знания по дисциплине, концентрируют внимание на наиболее сложных и важных вопросах. Во время лекционных занятий рекомендуется вести конспектирование учебного материала; обращать внимание на формулировки и категории, раскрывающие суть проблемы, явления или процесса; зафиксировать выводы и практические рекомендации. Подготовка к **лабораторным занятиям** включает ознакомление с планом лабораторного занятия; работу с конспектом лекций, выполнение домашнего задания, работу с учебной и учебно-методической литературой, научными изданиями и электронными образовательными ресурсами, рекомендованными рабочей программой дисциплины. Содержание **самостоятельной работы** определяется рабочей программой дисциплины, оценочными и методическими материалами, заданиями и указаниями преподавателя. Самостоятельная работа может осуществляться в аудиторной и внеаудиторной формах. Эффективным средством осуществления самостоятельной работы является электронная информационно-образовательная среда университета, которая обеспечивает доступ к образовательной программе, рабочей программе дисциплины, к электронным библиотечным системам, профессиональным базам данных и информационным справочным системам.

Периодичность проведения, формы текущего контроля успеваемости, система оценивания хода освоения дисциплин представлены в рабочей программе.

Раздел 6. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ И УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

6.1. Учебно-методическое обеспечение

№№ п/п	Список используемой литературы	Количество экземпляров печатных изданий, имеющих в библиотеке, или электронный адрес издания (ресурса) в сети Интернет
УЧЕБНЫЕ, УЧЕБНО-МЕТОДИЧЕСКИЕ И НАУЧНЫЕ ИЗДАНИЯ		
1.	Кошкин, Вениамин Васильевич. Системное программное обеспечение [Текст] : конспект лекций : [для студентов очной и очно-заочной форм обучения направлений	46 / https://portal.volgatech.net/books/Koshkin_sistemnoe_pro

	11.03.03 и 27.03.04] / В. В. Кошкин; М-во образования и науки Рос. Федерации, ФГБОУ ВПО "Поволж. гос. технол. ун-т". Йошкар-Ола: ПГТУ, 2015. - 195 с. ISBN 978-5-8158-1495-0. Экземпляры: всего 46.	grammnoe_obespechenie_2015.pdf
2.	Морохин, Дмитрий Витальевич. Основы теории трансляции [Текст] : лабораторный практикум : [по направлению подготовки бакалавров 09.03.01 "Информатика и вычислительная техника"] / Д. В. Морохин; М-во образования и науки Рос. Федерации, ФГБОУ [ВО] "Поволж. гос. технол. ун-т". Йошкар-Ола: ПГТУ, 2015. - 69 с. ISBN 978-5-8158-1587-2. Экземпляры: всего 33.	33
3.	Иванько, А. Ф. Системное программное обеспечение информационных мультимедиа-систем [Текст] / Иванько А. Ф., Иванько М. А. Санкт-Петербург: Лань, 2020. - 80 с. с. ISBN 978-5-8114-4927-9.	https://e.lanbook.com/book/139325
4.	Жулабова, Ф. Т. Системное программирование. Лабораторные работы [Электронный ресурс] : учебное пособие / Жулабова Ф. Т. Санкт-Петербург: Лань, 2020. - 208 с. ISBN 978-5-8114-4666-7.	https://e.lanbook.com/book/140772
5.	Вояковская, Н. Н. Разработка компиляторов [Электронный ресурс] / Вояковская Н. Н., Москаль А. Е., Булычев Д. Ю., Терехов А. А. 2-е изд. Москва: ИНТУИТ, 2016. - 374 с.	https://e.lanbook.com/book/100452
6.	Шорников, Ю. В. Теория языков и языковых процессоров [Электронный ресурс] : учебник для вузов / Шорников Ю. В. 2-е изд., перераб. и доп. Санкт-Петербург: Лань, 2024. - 292 с. ISBN 978-5-507-48427-0.	https://e.lanbook.com/book/380705
ПРОФЕССИОНАЛЬНЫЕ БАЗЫ ДАННЫХ И ИНФОРМАЦИОННЫЕ СПРАВОЧНЫЕ СИСТЕМЫ		
1.	Справочно-правовая система Консультант+	http://www.consultant.ru
2.	Профессиональные справочные системы Техэксперт	http://www.cntd.ru

6.2. Материально-техническая база и программное обеспечение

№№ п/п	Аудитории для проведения учебных занятий, самостоятельной работы и проведения государственной итоговой аттестации	Перечень основного оборудования	Программное обеспечение
1.	510 (III)	Экран настенный рулонный 200x200 см (1), Комплект учебной мебели (1)	Microsoft Windows Enterprise, Microsoft Office Standard, Microsoft Visio Professional, Microsoft Visual Studio Enterprise, Комплект ПО для решения основных пользовательских задач
2.	518 (III)	ПК 5 - ICL RAY P222.3 ,клавиат.,мышь.,монитор LG E2251T-BN (14), Комплект учебной	Microsoft Windows Enterprise, Microsoft Office Standard,

		мебели (1)	Microsoft Visio Professional, Microsoft Visual Studio Enterprise, Комплект ПО для решения основных пользовательских задач
3.	519 (III)	Монитор 17" BenQ FP 71G (1), Монитор TET 20" Samsung SIM 2043W (1), ОСЦИЛЛОГРАФ C1-75 (1), ПК ICL RAY H494.1 сист.блок,клавиат,мышь,монитор View Sonic VA2231 WLED WZ1218) (14), Сист. блок CE 331/256*2/PC 3200/80 Gb/FDD/DVD-ROM/КЛАВ+МЫШЬ+коврик (1),	Microsoft Windows Enterprise, Microsoft Office Standard, Microsoft Visio Professional, Microsoft Visual Studio Enterprise, Комплект ПО для решения основных пользовательских задач

Раздел 7. ФОРМЫ КОНТРОЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ/ ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

Критерии оценивания индикаторов достижения компетенций направлены на:

- усвоение теоретического материала (объем знаний, глубина усвоения), предусмотренного рабочей программой;
- умение излагать материал (четкость, грамотность изложения материала, точность и полнота воспроизведения учебного материала);
- умение применять теоретические знания при решении практических заданий.

Шкала оценивания представлена ниже.

Уровень сформированности элементов компетенции	Критерии оценивания	Шкала оценивания
Пороговый уровень	Обучающийся имеет знания основного материала, проявляет умение логично его излагать, но может допускать неточности в изложении материала, недостаточно правильные формулировки, испытывает затруднения в выполнении практических заданий	Зачтено

7.1. Текущий контроль успеваемости

Текущий контроль успеваемости обеспечивает оценивание хода освоения дисциплины (модуля) и производится с применением технологии рейтингового контроля в соответствии с технологической картой дисциплины. Порядок составления технологической карты и алгоритм проведения процедуры оценивания видов деятельности обучающихся, направленных на освоение знаний, умений, навыков и/или опыта деятельности, по накопительной системе в баллах устанавливается положением о системе РИТМ в ФГБОУ ВО «ПГТУ»

7.2. Промежуточная аттестация обучающихся

Промежуточная аттестация обучающихся направлена на оценивание результатов обучения по дисциплине (модулю) и проводится с использованием фондов оценочных средств.

Примеры типовых контрольных заданий из базы фонда оценочных средств по образовательной программе.

Билет к зачету № 1

по дисциплине «Системное программное обеспечение»

1. Как в общем случае осуществляется загрузка ОС после включения компьютера? Что такое системный и внесистемный загрузчики? Где они располагаются?
2. Какие типы грамматик выделяют по классификации Хомского? Как они между собой соотносятся?
3. Какой цели служит преобразование правил КС-грамматик? Всегда ли это преобразование ведет к упрощению правил?

Билет к зачету № 2

по дисциплине «Системное программное обеспечение»

1. Что такое «файловая система»? Что обеспечивает использование той или иной файловой системы? Какие файловые системы, используемые в ОС для ПК, вы знаете?
2. Опишите структуру магнитного диска (разбиение дисков на разделы). Сколько (и каких) разделов может быть на магнитном диске?
3. Как в общем случае осуществляется загрузка ОС после включения компьютера? Что такое системный и внесистемный загрузчики? Где они располагаются?

Билет к зачету № 3

по дисциплине «Системное программное обеспечение»

1. Объясните общие принципы файловой системы FAT. Что такое кластер, от чего зависит его размер?
2. Что такое трансляция, компиляция, транслятор, компилятор? Из каких процессов состоит компиляция?
3. Какую роль выполняет лексический анализ в процессе компиляции? Как могут быть связаны между собой лексический и синтаксический анализ?

Билет к зачету № 4

по дисциплине «Системное программное обеспечение»

1. Компилятор можно построить, не используя лексический анализатор. Объясните, почему все-таки большинство современных компиляторов используют фазу лексического анализа.
2. Почему в глобальной сети Интернет используются в основном языки программирования,

предусматривающие интерпретацию исходного кода? Какие проблемы проистекают из этого факта?

3. Почему для языка ассемблера, который имеет простую (часто — регулярную) грамматику, чаще всего используется двухпроходный компилятор? Что мешает свести компиляцию исходного кода на языке ассемблера в один проход?

Билет к зачету № 5

по дисциплине «Системное программное обеспечение»

1. Что такое таблица идентификаторов и для чего она предназначена? Какая информация может храниться в таблице идентификаторов?
2. Чем различаются таблица лексем и таблица идентификаторов? В какую из этих таблиц лексический анализатор не должен помещать ключевые слова, разделители и знаки операций?
3. В чем различия между деревом вывода и деревом операций, если и то и другое получается в результате синтаксического разбора?

Билет к зачету № 6

по дисциплине «Системное программное обеспечение»

1. Дайте определение цепочки, языка. Какие операции можно выполнять над цепочками символов? Что такое синтаксис и семантика языка?
2. Какие из перечисленных ниже тождеств являются истинными для двух произвольных цепочек символов, а какие нет?
3. Какие существуют методы задания языков? Почему метод перечисления всех допустимых цепочек языка не находит практического применения?

Билет к зачету № 7

по дисциплине «Системное программное обеспечение»

1. Какие дополнительные вопросы необходимо решить при задании языка программирования? Почему любой язык программирования не является чисто формальным языком?
2. Как решается вопрос о смысле исходной программы в современных компиляторах?
3. Что такое грамматика языка? Дайте определения грамматики.

Билет к зачету № 8

по дисциплине «Системное программное обеспечение»

1. Как выглядит описание грамматики в форме Бэкуса—Наура? Какие еще формы описания грамматик существуют?
2. Почему в форме Бэкуса—Наура практически невозможно построить грамматику для реального языка так, чтобы она не содержала рекурсивных правил?
3. Какие типы грамматик выделяют по классификации Хомского? Как они между собой соотносятся?

Перечень вопросов для проведения промежуточной аттестации

4. Что такое «файловая система»? Что обеспечивает использование той или иной файловой системы? Какие файловые системы, используемые в ОС для ПК, вы знаете?
5. Опишите структуру магнитного диска (разбиение дисков на разделы). Сколько (и каких) разделов может быть на магнитном диске?
6. Как в общем случае осуществляется загрузка ОС после включения компьютера? Что такое системный и внесистемный загрузчики? Где они располагаются?
7. Объясните общие принципы файловой системы FAT. Что такое кластер, от чего зависит его размер?
8. Что такое трансляция, компиляция, транслятор, компилятор? Из каких процессов состоит компиляция?
9. Какую роль выполняет лексический анализ в процессе компиляции? Как могут быть связаны между собой лексический и синтаксический анализ?
10. Компилятор можно построить, не используя лексический анализатор. Объясните, почему все-таки большинство современных компиляторов используют фазу лексического анализа.
11. Почему в глобальной сети Интернет используются в основном языки программирования, предусматривающие интерпретацию исходного кода? Какие проблемы проистекают из этого факта?
12. Почему для языка ассемблера, который имеет простую (часто — регулярную) грамматику, чаще всего используется двухпроходный компилятор? Что мешает свести компиляцию исходного кода на языке ассемблера в один проход?

13. Что такое таблица идентификаторов и для чего она предназначена? Какая информация может храниться в таблице идентификаторов?
14. Чем различаются таблица лексем и таблица идентификаторов? В какую из этих таблиц лексический анализатор не должен помещать ключевые слова, разделители и знаки операций?
15. В чем различия между деревом вывода и деревом операций, если и то и другое получается в результате синтаксического разбора?
16. Дайте определение цепочки, языка. Какие операции можно выполнять над цепочками символов? Что такое синтаксис и семантика языка?
17. Какие из перечисленных ниже тождеств являются истинными для двух произвольных цепочек символов, а какие нет?
18. Какие существуют методы задания языков? Почему метод перечисления всех допустимых цепочек языка не находит практического применения?
19. Какие дополнительные вопросы необходимо решить при задании языка программирования? Почему любой язык программирования не является чисто формальным языком?
20. Как решается вопрос о смысле исходной программы в современных компиляторах?
21. Что такое грамматика языка? Дайте определения грамматики.
22. Как выглядит описание грамматики в форме Бэкуса—Наура? Какие еще формы описания грамматик существуют?
23. Почему в форме Бэкуса—Наура практически невозможно построить грамматику для реального языка так, чтобы она не содержала рекурсивных правил?
24. Какие типы грамматик выделяют по классификации Хомского? Как они между собой соотносятся?
25. Какие типы языков выделяют по классификации Хомского? Как классификация языков соотносится с классификацией грамматик?
26. Дайте определения выводимости цепочки, непосредственной выводимости нетривиальной выводимости, длины вывода.
27. Что такое сентенциальная форма грамматики? Если язык, заданный грамматикой, представляет собой множество ее конечных сентенциальных форм, что представляет собой множество всех сентенциальных форм грамматики?4;
28. Что такое левосторонний и правосторонний выводы? Можно ли построить еще какие-нибудь варианты цепочек вывода?
29. Как классифицируются распознаватели? Как их классификация соотносится с классификацией языков и грамматик?
30. Кто (или что) для любого языка программирования выступает в роли генератора цепочек языка? Кто (или что) выступает в роли распознавателя цепочек? Как решается задача разбора?
31. В чем заключается отличие автоматных грамматик от других регулярных грамматик? Всякая ли регулярная грамматика является автоматной? Всякая ли регулярная грамматика может

быть преобразована к автоматному виду?

32. Всегда ли недетерминированный КА может быть преобразован к детерминированному виду (если нет — то в каких случаях)?
33. Можно ли для языка, заданного левوليнейной грамматикой, построить праволинейную грамматику, задающую эквивалентный язык?
34. Всякая ли регулярная грамматика является однозначной? Если нет, то приведите пример неоднозначной регулярной грамматики.
35. Алфавит магазинных символов МП-автомата включает в себя алфавит входных символов автомата. Почему?
36. Если проблема однозначности неразрешима для произвольной КС-грамматики, то почему можно утверждать, что некоторые КС-грамматики являются однозначными? Всегда ли должны быть однозначными грамматики синтаксических конструкций языков программирования?
37. Какой цели служит преобразование правил КС-грамматик? Всегда ли это преобразование ведет к упрощению правил?
38. Почему при преобразовании КС-грамматики к приведенному виду сначала необходимо удалить бесплодные символы, а потом — недостижимые символы?
39. Какие преимущества дает запись правил грамматики в нормальной форме Хомского? Ведет ли она к упрощению правил грамматики? Где используются правила грамматики в нормальной форме Хомского?
40. В чем главное преимущество линейных распознавателей для КС-языков перед всеми другими видами распознавателей? Какие существуют сложности в построении линейных распознавателей для КС-языков?
41. В чем особенности реализации алгоритма типа «сдвиг-свертка» для грамматик простого и операторного предшествования? Какие ограничения эти особенности накладывают на правила грамматики?
42. Справедливы ли следующие утверждения:
43. любой язык программирования является контекстно-свободным языком;
44. синтаксис любого языка программирования может быть описан с помощью контекстно-свободной грамматики;
45. любой язык программирования является контекстно-зависимым языком;
46. семантика любого языка программирования может быть описана с помощью контекстно-свободной грамматики?
47. Какие задачи в компиляторе решает семантический анализ? Можно ли построить компилятор без семантического анализатора? Если да, то какие условия должны при этом выполняться?
48. Цепочка символов, принадлежащая любому языку программирования, может быть распознана с помощью распознавателя для контекстно-зависимых языков. При этом не будет требоваться дополнительный семантический анализ цепочки. Почему такой подход не применяется в компиляторах на практике?
49. Почему в большинстве компиляторов помимо генерации результирующего объектного кода выполняется еще и оптимизация его? Можно ли построить компилятор, исключив фазу

оптимизации кода?

50. От чего зависит эффективность объектного кода, построенного компилятором? Влияют ли на эффективность результирующего кода синтаксис и семантика исходного языка программирования?
51. Сравните между собой основные способы внутреннего представления программ. На каких этапах компиляции лучше всего использовать каждый из этих способов? Какие (или какой) из способов внутреннего представления программы обязательно должен уметь обрабатывать компилятор?
52. Почему имеются трудности с оптимизацией выражений, представленных в форме обратной польской записи?
53. Какой из двух основных методов оптимизации: машинно-зависимый или машинно-независимый может порождать более эффективный результирующий код? Как сочетаются эти два метода оптимизации в компиляторе?
54. Почему линейные участки легче поддаются оптимизации, чем все другие части внутреннего представления программы и объектного кода?
55. Почему можно утверждать, что исключение из исходного языка операций с указателями и адресами значительно усиливает возможности компилятора по оптимизации кода?
56. Какое значение будет иметь переменная *i* после выполнения следующего оператора языка C:

```
if (++i>10 || i++>20 && 1++<50 || i++<0) i--;
```

если переменная *i* имеет перед его выполнением следующие значения:

9 11 21 0 -2?

57. Какие варианты распараллеливания выполнения операций может использовать компилятор при порождении объектного кода, необходимого для вычисления выражения

« $A*B*C + A*D/E - B/E - F$ »?