

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
Факультет физико-технический

УТВЕРЖДАЮ  
Проректор по учебной работе,  
качеству образования - первый  
проректор  
Т.А. Хагуров  
подпись  
« 23 » мая 2023 г. 05



**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)**  
**Б1.В.ДВ.01.02 Продвинутые аспекты программирования и настройки**  
**производительности: PL/SQL**

Направление подготовки/специальность 09.03.02 Информационные системы и технологии

Направленность (профиль)/ специализация Аналитические информационные системы

Форма обучения заочна

Квалификация бакалавр

Краснодар 2023

Рабочая программа дисциплины Б1.В.ДВ.01.02 Продвинутые аспекты программирования и настройки производительности: PL/SQL составлена в соответствии с федеральным государственным образовательным стандартом высшего образования (ФГОС ВО) по направлению подготовки/специальности 09.03.02 Информационные системы и технологии

Программу составил (и):


В.В. Лежнев, доцент кафедры теор. физики и комп. технологий, кандидат физ.- мат. наук



подпись

Рабочая программа дисциплины Б1.В.ДВ.01.02 Продвинутые аспекты программирования и настройки производительности: PL/SQL утверждена на заседании кафедры теоретической физики и компьютерных технологий протокол № от «12» апреля 2023 г.

Заведующий кафедрой (выпускающей) Лебедев К.А.



подпись

Утверждена на заседании учебно-методической комиссии физико-технического факультета

протокол № от « » апреля 2023 г.

Председатель УМК факультета Богатов Н.М.



подпись

Рецензенты:

М.С. Коваленко, кандидат физико-математических наук, доцент кафедры физики и информационных систем

Л.Р. Григорян, генеральный директор ООО НПФ «Мезон»  
кандидат физико-математических наук

**Объем трудоемкости:** 4 зачетных единицы (144 часов (в 7 семестре), из них – 46 часов аудиторной нагрузки: лекционных 12 ч., лабораторных 22 ч. практических 12 ч.; 65 часов самостоятельной работы)

**Цель дисциплины «Продвинутые аспекты программирования и настройки производительности: PL/SQL»** создание эффективных пакетов и программных единиц PL/SQL написание кода, взаимодействующего с другими приложениями и операционной системой создание приложений, использующих коллекции создание оптимальных с точки зрения производительности программ, применение механизмов детального контроля доступа написание кода, взаимодействующего с большими объектами (LOBs), и использование SecureFile LOBs

#### **Задачи дисциплины:**

1. Использовать внешние процедуры, содержащиеся в DLL, и интегрировать в приложение PL/SQL.
2. Использовать динамический SQL (dynamic SQL) для расширения функциональности программ и повышения их гибкости, как и встроенные в язык конструкции (native dynamic SQL), так и процедуры пакета DBMS\_SQL. Формирование навыков владения моделями и средствами разработки архитектуры информационных систем.

#### **Место дисциплины в структуре образовательной программы**

Дисциплина «Продвинутые аспекты программирования и настройки производительности: PL/SQL» относится к базовому циклу дисциплин. Для освоения данной дисциплины необходимы знания, умения и компетенции, полученные обучающимися при изучении курсов «Информатика», «Операционные системы».

Дисциплина обеспечивает изучение дисциплин профессионального цикла. Рабочая программа дисциплины «Продвинутые аспекты программирования и настройки производительности: PL/SQL» имеет трудоемкость равную 4 зачетным единицам.

В результате изучения дисциплины студент должен знать: принципы организации и возможности инструментальных программных средств для решения задач в области создания и редактирования системного ПО; основные структуры данных (списки, множества и т.п.), методы их обработки и способы реализации; методы и технологии структурного и объектно-ориентированного программирования; основные виды ошибок и способы тестирования программного обеспечения; методы разработки алгоритмов и способы проверки их корректности; уметь: использовать на практике специализированные инструментальные средства для создания компонентов прикладного программного обеспечения; определять эффективные структуры данных для решения задачи структурировать информационные проблемы; применять профессио-

нальные знания для создания эффективных алгоритмов; реализовывать построенные алгоритмы в виде программ на конкретном языке; выбирать подходящую парадигму и язык программирования для конкретных задач; осуществлять проверку корректности создаваемых алгоритмов и программ; документировать результаты проектирования и программирования программных продуктов.

владеть: навыками создания программ и программных систем в интегрированных средах программирования; навыками разработки эффективных структур данных; навыками анализа задач и создания их формализованного представления; навыками выбора подходящей парадигмы и язык программирования для конкретных задач; навыками тестирования и проверки эффективности программ и программных систем.

### Требования к уровню освоения дисциплины

Изучение данной учебной дисциплины направлено на формирование у обучающихся следующих компетенций:

Код и наименование индикатора*	Результаты обучения по дисциплине
ПК-4	Способность создания (модификации) и сопровождения информационных систем (ИС), автоматизирующих задачи организационного управления и бизнес-процессы в организациях различных форм собственности с целью повышения эффективности деятельности организаций - пользователей ИС

### Содержание дисциплины:

Распределение видов учебной работы и их трудоемкости по разделам дисциплины.

№	Наименование разделов (тем)	Количество часов				
		Всего	Аудиторная работа			Внеаудиторная работа
			Л	ПЗ	ЛР	
7 семестр						
1.	Обзор концепций программирования на PL/SQL	18	1	1	2	8
2.	Дизайн кода PL/SQL	18	1	1	2	8
3.	Продвинутое программирование: Динамический SQL.	18	2	2	2	8
4.	Использование коллекций	18	2	2	2	8
5.	Манипулирование большими объектами	18	2	2	4	8
6.	Увеличение производительности за счет кэширования	18	2	2	4	9
7.	Анализ кода PL/SQL	18	1	1	4	8

8.	Продвинутое программирование: использование кол-лекций.	18	1	1	2	8
	<i>ИТОГО по разделам дисциплины</i>	144	12	12	22	65
	Контроль самостоятельной работы (КСР)	6				
	Промежуточная аттестация (ИКР)	0,3				
	Подготовка к текущему контролю	26,7				
	Общая трудоемкость по дисциплине	177				

**Курсовые работы:** не предусмотрены

**Форма проведения аттестации по дисциплине:** экзамен (7 семестр)

## СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

### Модуль 1. Основы алгоритмизации

#### Тема 1.1 Специфика решения задачи с использованием компьютера

Рассматривается роль моделирования при решении сложных задач, приводится типовая совокупность моделей, описывающих состав и поведение сложного объекта, обосновывается применение компьютерной техники при решении проблем, связанных со сложным объектом. Анализируется устройство компьютера как инструмента решения задач, акцентируется внимание на трех составляющих компьютера: процессоре, оперативной памяти, устройствах ввода-вывода. Подчеркивается основной принцип действия компьютера – хранение программ и данных в адресном пространстве оперативной памяти и обращение к ним процессора в ходе решения задачи. Приводится пример создания совокупности моделей (концептуальной, формальной, математической, алгоритмической, программной) при решении задачи с использованием компьютера.

#### Тема 1.2 Понятие алгоритма

Приводится два определения алгоритма: как фундаментального универсального понятия и как вычислительного процесса. Перечисляются основные свойства алгоритмов и способы их записи в виде псевдокодов и блок-схем. Поясняется понятие «исполнение алгоритма» как практическая реализация действий по получению результата для конкретных значений данных и ее запись в табличном

виде. Рассматривается принцип структурной алгоритмизации как основы для технологии структурного программирования. Вводится базовый набор алгоритмических структур (линейная, ветвящаяся, циклическая), приводятся примеры записи базовых структур формализмом блок-схем.

#### Тема 1.3 Примеры классических алгоритмов

Поясняется назначение и широкое использование в алгоритмах переменных-счетчиков и аккумуляторов для получения итоговых значений (подсчет ите-

раций цикла, подсчет суммы с накоплением). Анализируется алгоритм перестановки значений двух переменных, лежащий в основе алгоритмов сортировки.

Рассматриваются алгоритмы последовательного и двоичного поиска числа в массиве данных.

## Модуль 2. «Основы программирования»

Тема 2.1 Эволюция программирования как деятельности  
Приводится ретроспектива развития средств вычислительной техники и логических основ ее функционирования. Рассматривается развитие парадигм программирования как совокупности идей и понятий, определяющей стиль написания программ. Приводится классификация языков программирования по их уровням и принадлежности к парадигмам. Программирование как вид деятельности рассматривается как наука, как искусство и как ремесло.

### Тема 2.2 Инструменты программирования

Вводятся основные понятия программирования: синтаксис и семантика языка программирования, текст программы на языке программирования, трансляция как перевод текста на машинный язык, тестирование и отладка программы. Отмечаются функции транслятора в процессе создания программы и приводятся их разновидности. Рассматривается структура инструментальной среды для создания программ (системы программирования) и роль ее составляющих. Приводятся наиболее популярные системы программирования.

### Тема 2.3 Проектирование и внедрение программ

Рассматриваются два этапа создания небольших и средних по объему кода программ: системный анализ и запись алгоритма на языке программирования.

Разработка крупного проекта информационной системы представлена восемью этапами, 50-90% объема проекта занимает этап внедрения. К методам маркетинга программного обеспечения отнесено коммерческое, условно-бесплатное и бесплатное распространение программного продукта.

## Модуль 3. Технология алгоритмического программирования

### Тема 3.1 Понятия алгоритмического программирования

Рассмотрение темы базируется на понимании сущности данных, их классификации для целей программирования. Различаются базовые (простые) и сложные (структурированные) типы данных. Поясняются понятия переменной и константы и их назначение. Программа представляется как совокупность операторов, предназначенных для целенаправленного преобразо-

вания данных. Рассмотрены виды операторов (арифметические, логические, управления ходом программы, ввода-вывода) на примере трех языков программирования (Basic, Pascal и C++), проиллюстрированы структура типовой программы, способы описания и инициализации переменных, группировка операторов, способы задания комментариев. Подчеркивается сходство операторов, реализующих базовые алгоритмические структуры, в различных языках программирования.

### Тема 3.2 Принципы структурного программирования

Структурное программирование рассматривается как программная реализация структурной алгоритмизации. Декомпозиция алгоритмов реализуется совокупностью подпрограмм, вызываемых из главной программы. Вводятся понятия «процедуры» и «функции» как разновидности подпрограмм. Описывается способ взаимодействия подпрограмм посредством передачи параметров при решении общей задачи. На примере трех языков программирования показаны способы описания процедур и функций в теле основной программы.

### Модуль 4 «Технология событийного программирования»

#### Тема 4.1 Основы событийного программирования

Понятие «визуальное программирование» ассоциируется с работой в графической среде программирования, а результат программирования - с windows-

приложениями в виде окон (экранных форм) с набором элементов управления.

Действия пользователя рассматриваются как события, на которые реагирует приложение. Разработка таких приложений называется событийным программированием и заключается в конструировании экранных форм, определении

значений свойств элементов управления экранной формы, создании программного

кода как совокупность методов обработки событий.

#### Тема 4.2 Объектно-ориентированное программирование

Представляется как самая современная технология программирования, реализующая соответствующую парадигму и обеспеченная визуальными средствами конструирования приложений на базе объектных репозиториях.

Поясняются базовые понятия объекта и класса, дается их интерпретация как структурированного типа данных, включающего данные (свойства объекта) и подпрограммы (методы реакции объекта на события). Проводится сравнение с другими структурированными типами данных. На примере языков Pascal и C++ иллюстрируются способы описания класса, инициализации объекта как экземпляра класса. На примерах поясняются три основные концепции объектно-

ориентированного программирования: инкапсуляция, наследование, полиморфизм.

## 12. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине и методические указания для обучающихся по освоению дисциплины

### - Методические указания к лекционным занятиям

Лекция является основной формой обучения в высшем учебном заведении. В ходе лекционного курса проводится изложение современных материалов в области программирования. В процессе обучения могут быть использованы видео материалы. Копии видео файлов доступны для повторного просмотра при самостоятельной работе.

В тетради для конспектирования лекций необходимо иметь поля, где по ходу конспектирования делаются необходимые пометки. Записи должны быть избирательными, полностью следует записывать только определения. В конспекте применяется применять сокращение слов, что ускоряет запись. Вопросы, возникающие в ходе лекции, рекомендуется записывать на полях и после окончания лекции обратиться за разъяснением к преподавателю. Необходимо активно работать с конспектом лекции: после окончания лекции рекомендуется перечитать свои записи, внести поправки и дополнения на полях. Конспекты лекций следует использовать при подготовке к семинарам, при подготовке к экзамену, контрольным вопросам.

### - Методические указания к практическим занятиям

Практические занятия по курсу «Технологии программирования» имеют целью закрепить у студентов навыки разработки программного обеспечения. Прохождение всего цикла практических занятий является условием допуска студента к экзамену.

Студент должен вести активную познавательную работу. Важно научиться включать вновь получаемую информацию в систему уже имеющихся знаний. Необходимо также анализировать материал для выделения общего в частном, и наоборот, частного в общем.

## Примерный перечень вопросов к зачету

1. Роль моделирования при решении сложных задач
2. Типовая совокупность моделей, описывающая состав и поведение сложного объекта.
3. Применение компьютерной техники при решении проблем, связанных со сложным объектом.
4. Анализ устройства компьютера как инструмента решения задач (процессор, оперативная память, устройства ввода-вывода)
5. Определения алгоритма: как фундаментального универсального понятия и как вычислительного процесса.



6. Основные свойства алгоритмов и способы их записи в виде псевдокодов и блок-схем.
7. Понятие «исполнение алгоритма» как практическая реализация действий по получению результата для конкретных значений данных и ее запись в табличном виде.
8. Принцип структурной алгоритмизации как основы для технологии структурного программирования.
9. Примеры записи базовых структур формализмом блок-схем.
10. Назначение и широкое использование в алгоритмах переменных-счетчиков и аккумуляторов для получения итоговых значений (подсчет итераций цикла, подсчет суммы с накоплением).
11. Алгоритм перестановки значений двух переменных, лежащий в основе алгоритмов сортировки.
12. Алгоритмы последовательного и двоичного поиска числа в массиве данных.
13. Ретроспектива развития средств вычислительной техники и логических основ ее функционирования.
14. Развитие парадигм программирования как совокупности идей и понятий, определяющей стиль написания программ.
15. Классификация языков программирования по их уровням и принадлежности к парадигмам.
16. Программирование как вид деятельности рассматривается как наука, как искусство и как ремесло.
17. Основные понятия программирования: синтаксис и семантика языка программирования
18. Функции транслятора в процессе создания программы и приводятся их разновидности.
19. Структура инструментальной среды для создания программ (системы программирования) и роль ее составляющих. Наиболее популярные системы программирования.
20. Этапы создания небольших и средних по объему кода программ: системный анализ и запись алгоритма на языке программирования.
21. Понятия алгоритмического программирования
22. Виды операторов (арифметические, логические, управления ходом программы, ввода-вывода) на примере трех языков программирования (Basic, Pascal и C++)
23. Принципы структурного программирования
24. Декомпозиция алгоритмов.
25. Понятия «процедуры» и «функции» как разновидности подпрограмм. Описывается способ взаимодействия подпрограмм посредством передачи параметров при решении общей задачи.
26. Основы событийного программирования

Автор к.ф.-м.н. Лежнев В.В.