

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное учреждение
высшего профессионального образования
"Казанский (Приволжский) федеральный университет"
Институт вычислительной математики и информационных технологий



подписано электронно-цифровой подписью

Программа дисциплины

Дополнительные главы информатики Б2.В.1

Направление подготовки: 010400.62 - Прикладная математика и информатика

Профиль подготовки: Математическая кибернетика

Квалификация выпускника: бакалавр

Форма обучения: очное

Язык обучения: русский

Автор(ы):

Ахтямов Р.Б. , Бухараев Н.Р. , Самитов Р.К.

Рецензент(ы):

Еникеев А.И.

СОГЛАСОВАНО:

Заведующий(ая) кафедрой: Аблаев Ф. М.

Протокол заседания кафедры No ____ от " ____ " _____ 201__ г

Учебно-методическая комиссия Института вычислительной математики и информационных технологий:

Протокол заседания УМК No ____ от " ____ " _____ 201__ г

Регистрационный No 9160614

Казань
2014

Содержание

1. Цели освоения дисциплины
2. Место дисциплины в структуре основной образовательной программы
3. Компетенции обучающегося, формируемые в результате освоения дисциплины /модуля
4. Структура и содержание дисциплины/ модуля
5. Образовательные технологии, включая интерактивные формы обучения
6. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины и учебно-методическое обеспечение самостоятельной работы студентов
7. Литература
8. Интернет-ресурсы
9. Материально-техническое обеспечение дисциплины/модуля согласно утвержденному учебному плану

Программу дисциплины разработал(а)(и) доцент, к.н. (доцент) Ахтямов Р.Б. кафедра теоретической кибернетики отделение фундаментальной информатики и информационных технологий , Raouf.Akhiamov@kpfu.ru ; доцент, к.н. (доцент) Бухараев Н.Р. кафедра технологий программирования отделение фундаментальной информатики и информационных технологий , Naille.Boukharaev@kpfu.ru ; Самитов Р.К.

1. Цели освоения дисциплины

Цель курса - ввести в круг понятий и задач информатики, связанных с проблемами информационного моделирования объектов предметной области средствами языков программирования и обработки данных с помощью вычислительных машин. Задача курса состоит в выработке у студентов навыков использования структур данных и методов разработки алгоритмов на примере классических алгоритмов обработки данных, ввести понятия о формальном представлении алгоритмов, их сложности и об ЭВМ как исполнителях алгоритмов.

2. Место дисциплины в структуре основной образовательной программы высшего профессионального образования

Данная учебная дисциплина включена в раздел " Б2.В.1 Общепрофессиональный" основной образовательной программы 010400.62 Прикладная математика и информатика и относится к вариативной части. Осваивается на 2 курсе, 3 семестр.

Данная дисциплина относится к общепрофессиональным дисциплинам.

Читается на 2 курсе в 3 семестре для студентов обучающихся по направлению "Прикладная математика и информатика".

Изучение основывается на результатах изучения дисциплин "Основы информатики".

3. Компетенции обучающегося, формируемые в результате освоения дисциплины /модуля

В результате освоения дисциплины формируются следующие компетенции:

Шифр компетенции	Расшифровка приобретаемой компетенции
ПК-1 (профессиональные компетенции)	способность демонстрации общенаучных базовых знаний естественных наук, математики и информатики, понимание основных фактов, концепций, принципов теорий, связанных с прикладной математикой и информатикой;
ПК-2 (профессиональные компетенции)	способность приобретать новые научные и профессиональные знания, используя современные образовательные и информационные технологии;
ПК-6 (профессиональные компетенции)	способность осуществлять целенаправленный поиск информации о новейших научных и технологических достижениях в сети Интернет и из других источников;
ПК-8 (профессиональные компетенции)	способность формировать суждения о значении и последствиях своей профессиональной деятельности с учетом социальных, профессиональных и этических позиций;
ПК-9 (профессиональные компетенции)	способность решать задачи производственной и технологической деятельности на профессиональном уровне, включая: разработку алгоритмических и программных решений в области системного и прикладного программирования;
ОК-12 (общекультурные компетенции)	способность работать с информацией в глобальных компьютерных сетях.

Шифр компетенции	Расшифровка приобретаемой компетенции
ПК-10 (профессиональные компетенции)	способность применять в профессиональной деятельности современные языки программирования и языки баз данных, операционные системы, электронные библиотеки и пакеты программ, сетевые технологии;

В результате освоения дисциплины студент:

1. должен знать:

основные понятия и методы современного этапа развития разработки программного обеспечения

2. должен уметь:

применять освоенные понятия и методы в профессиональной деятельности

3. должен владеть:

базовыми знаниями и практическими навыками современного этапа методов разработки программного обеспечения

4. должен демонстрировать способность и готовность:

основные принципы технологии разработки программ.

4. должен демонстрировать способность и готовность:

ориентироваться в составе, особенностях, преимуществах и недостатках основных парадигм (стилей) программирования.

4. должен демонстрировать способность и готовность:

теоретическими знаниями о концепции типа данных и механизмах абстракции данных и процессов, о методах разработки алгоритмов решения задач и основах анализа алгоритмов.

4. должен демонстрировать способность и готовность:

навыки алгоритмизации задач и использования механизмов абстракции в разработке программ.

4. Структура и содержание дисциплины/ модуля

Общая трудоемкость дисциплины составляет 4 зачетных(ые) единиц(ы) 144 часа(ов).

Форма промежуточного контроля дисциплины экзамен в 3 семестре.

Суммарно по дисциплине можно получить 100 баллов, из них текущая работа оценивается в 50 баллов, итоговая форма контроля - в 50 баллов. Минимальное количество для допуска к зачету 28 баллов.

86 баллов и более - "отлично" (отл.);

71-85 баллов - "хорошо" (хор.);

55-70 баллов - "удовлетворительно" (удов.);

54 балла и менее - "неудовлетворительно" (неуд.).

4.1 Структура и содержание аудиторной работы по дисциплине/ модулю

Тематический план дисциплины/модуля

N	Раздел Дисциплины/ Модуля	Семестр	Неделя семестра	Виды и часы аудиторной работы, их трудоемкость (в часах)			Текущие формы контроля
				Лекции	Практические занятия	Лабораторные работы	
1.	Тема 1. ОБЪЕКТНАЯ МОДЕЛЬ В ПРОГРАММИРОВАНИИ.	3		9	0	0	устный опрос
2.	Тема 2. ЛОГИКО-АЛГЕБРАИЧЕСКАЯ МОДЕЛЬ В ПРОГРАММИРОВАНИИ.	3		9	0	0	устный опрос
3.	Тема 3. ЯЗЫК СТРУКТУРИРОВАННЫХ ЗАПРОСОВ SQL.	3		9	0	0	контрольная работа
4.	Тема 4. МАШИННО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ.	3		9	0	0	устный опрос
5.	Тема 5. ТЕОРЕТИЧЕСКАЯ И ПРАКТИЧЕСКАЯ РЕАЛИЗУЕМОСТЬ АЛГОРИТМОВ.	3		9	0	0	контрольная работа
6.	Тема 6. ВЫЧИСЛИТЕЛЬНАЯ СЛОЖНОСТЬ АЛГОРИТМОВ	3		9	0	0	устный опрос
	Тема . Итоговая форма контроля	3		0	0	0	экзамен
	Итого			54	0	0	

4.2 Содержание дисциплины

Тема 1. ОБЪЕКТНАЯ МОДЕЛЬ В ПРОГРАММИРОВАНИИ.

лекционное занятие (9 часа(ов)):

Понятие модуля. Локальные и глобальные, инкапсулированные и интерфейсные языковые объекты. Описания, области действия и правила видимости. Понятие объектно-ориентированного анализа. Классы и объекты. Поля и свойства, методы и события, конструкторы и деструкторы. Базовая семантика классов как абстрактных типов данных. Инкапсуляция, наследование и полиморфизм.

Тема 2. ЛОГИКО-АЛГЕБРАИЧЕСКАЯ МОДЕЛЬ В ПРОГРАММИРОВАНИИ.

лекционное занятие (9 часа(ов)):

Базы данных как аппарат информационного моделирования. Модель данных как расширение концепции абстрактного типа. Основные понятия реляционной модели данных ? таблицы (поля, строки, колонки и их типы), ключи таблицы, отношения между таблицами, ограничения целостности (условия правильности) базы данных и операции манипулирования данными.

Тема 3. ЯЗЫК СТРУКТУРИРОВАННЫХ ЗАПРОСОВ SQL.

лекционное занятие (9 часа(ов)):

Средства определения данных. Базовые типы данных. Оператор CREATE TABLE, определение типа колонок, первичного ключа и ограничений целостности по строкам, колонкам и межтабличным связям. Средства манипулирования данными. Базовые предикаты. Операторы SELECT, INSERT, DELETE и UPDATE. Объединение и соединение таблиц, группировка и групповые (агрегатные) функции, предикаты с подзапросами.

Тема 4. МАШИННО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ.

лекционное занятие (9 часа(ов)):

Базовая модель вычислителя, оператор перехода и помеченные операторы. Декомпозиция выражений, структур данных и структур управления. Представление данных базового типа, ссылочное именование (адресация) данных и действий-команд. Структура вычислительной машины и ход процессов.

Тема 5. ТЕОРЕТИЧЕСКАЯ И ПРАКТИЧЕСКАЯ РЕАЛИЗУЕМОСТЬ АЛГОРИТМОВ.

лекционное занятие (9 часа(ов)):

Понятие об алгоритмической полноте и алгоритмической неразрешимости. Фундаментальные примеры неразрешимых проблем.

Тема 6. ВЫЧИСЛИТЕЛЬНАЯ СЛОЖНОСТЬ АЛГОРИТМОВ

лекционное занятие (9 часа(ов)):

Верхние, средние и нижние (асимптотические) оценки. Анализ сложности на примере классических алгоритмов сортировки и поиска. P и NP классы сложности.

4.3 Структура и содержание самостоятельной работы дисциплины (модуля)

N	Раздел Дисциплины	Семестр	Неделя семестра	Виды самостоятельной работы студентов	Трудоемкость (в часах)	Формы контроля самостоятельной работы
1.	Тема 1. ОБЪЕКТНАЯ МОДЕЛЬ В ПРОГРАММИРОВАНИИ.	3		подготовка к устному опросу	9	устный опрос
2.	Тема 2. ЛОГИКО-АЛГЕБРАИЧЕСКАЯ МОДЕЛЬ В ПРОГРАММИРОВАНИИ.	3		подготовка к устному опросу	9	устный опрос
3.	Тема 3. ЯЗЫК СТРУКТУРИРОВАННЫХ ЗАПРОСОВ SQL.	3		подготовка к контрольной работе	4	контрольная работа
				подготовка к устному опросу	5	устный опрос
4.	Тема 4. МАШИННО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ.	3		подготовка к устному опросу	9	устный опрос
5.	Тема 5. ТЕОРЕТИЧЕСКАЯ И ПРАКТИЧЕСКАЯ РЕАЛИЗУЕМОСТЬ АЛГОРИТМОВ.	3		подготовка к контрольной работе	4	контрольная работа
				подготовка к устному опросу	5	устный опрос
6.	Тема 6. ВЫЧИСЛИТЕЛЬНАЯ СЛОЖНОСТЬ АЛГОРИТМОВ	3		подготовка к устному опросу	9	устный опрос
	Итого				54	

5. Образовательные технологии, включая интерактивные формы обучения

Обучение происходит в форме лекционных занятий, а также самостоятельной работы студентов.

Теоретический материал излагается на лекциях. Причем конспект лекций, который остается у студента в результате прослушивания лекции не может заменить учебник. Его цель - формулировка основных утверждений и определений. Прослушав лекцию, полезно ознакомиться с более подробным изложением материала в учебнике. Список литературы разделен на две категории: необходимый для сдачи экзамена минимум и дополнительная литература.

Самостоятельная работа предполагает выполнение домашних работ. Практические задания, выполненные в аудитории, предназначены для указания общих методов решения задач определенного типа. Закрепить навыки можно лишь в результате самостоятельной работы. Кроме того, самостоятельная работа включает подготовку к экзамену. При подготовке к сдаче экзамена весь объем работы рекомендуется распределять равномерно по дням, отведенным для подготовки к экзамену, контролировать каждый день выполнения работы. Лучше, если можно перевыполнить план. Тогда всегда будет резерв времени.

6. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины и учебно-методическое обеспечение самостоятельной работы студентов

Тема 1. ОБЪЕКТНАЯ МОДЕЛЬ В ПРОГРАММИРОВАНИИ.

устный опрос , примерные вопросы:

Понятие модуля. Локальные и глобальные, инкапсулированные и интерфейсные языковые объекты. Описания, области действия и правила видимости. Понятие объектно-ориентированного анализа. Классы и объекты. Поля и свойства, методы и события, конструкторы и деструкторы. Базовая семантика классов как абстрактных типов данных. Инкапсуляция, наследование и полиморфизм.

Тема 2. ЛОГИКО-АЛГЕБРАИЧЕСКАЯ МОДЕЛЬ В ПРОГРАММИРОВАНИИ.

устный опрос , примерные вопросы:

Базы данных как аппарат информационного моделирования. Модель данных как расширение концепции абстрактного типа. Основные понятия реляционной модели данных ? таблицы (поля, строки, колонки и их типы), ключи таблицы, отношения между таблицами, ограничения целостности (условия правильности) базы данных и операции манипулирования данными.

Тема 3. ЯЗЫК СТРУКТУРИРОВАННЫХ ЗАПРОСОВ SQL.

контрольная работа , примерные вопросы:

Выбрать продавцов, которые моложе по меньшей мере одного из своих покупателей. Выбрать продавцов, которые моложе (в среднем) всех своих покупателей.

устный опрос , примерные вопросы:

Средства определения данных. Базовые типы данных. Оператор CREATE TABLE, определение типа колонок, первичного ключа и ограничений целостности по строкам, колонкам и межтабличным связям. Средства манипулирования данными. Базовые предикаты. Операторы SELECT, INSERT, DELETE и UPDATE. Объединение и соединение таблиц, группировка и групповые (агрегатные) функции, предикаты с подзапросами.

Тема 4. МАШИННО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ.

устный опрос , примерные вопросы:

Базовая модель вычислителя, оператор перехода и помеченные операторы. Декомпозиция выражений, структур данных и структур управления. Представление данных базового типа, ссылочное именование (адресация) данных и действий-команд. Структура вычислительной машины и ход процессов.

Тема 5. ТЕОРЕТИЧЕСКАЯ И ПРАКТИЧЕСКАЯ РЕАЛИЗУЕМОСТЬ АЛГОРИТМОВ.

контрольная работа , примерные вопросы:

Задать класс Vector. Определить поля (private): размерность указатель на массив. Методы (public) конструктор с одним параметром размерность конструктор с двумя параметрами размерность и указатель на массив деструктор показать объект перегрузка оператора+ - сложение векторов перегрузка оператора - - разность векторов перегрузка оператора * -умножение вектора на действительное число справа В программе реализовать методы класса. устный опрос , примерные вопросы:

Понятие об алгоритмической полноте и алгоритмической неразрешимости. Фундаментальные примеры неразрешимых проблем.

Тема 6. ВЫЧИСЛИТЕЛЬНАЯ СЛОЖНОСТЬ АЛГОРИТМОВ

устный опрос , примерные вопросы:

Верхние, средние и нижние (асимптотические) оценки. Анализ сложности на примере классических алгоритмов сортировки и поиска. P и NP классы сложности.

Тема . Итоговая форма контроля

Примерные вопросы к экзамену:

По данной дисциплине предусмотрено проведение экзамена.

БИЛЕТЫ К ЭКЗАМЕНАМ

Информатика, З. 1

Введение в реляционные базы данных.

DB1. Объекты и взаимосвязи как отношения. ER-диаграммы. Классификация отношений. Родительская и дочерняя таблицы.

DB2 Навигационный и реляционный подход к БД. Реализация связей ключевыми выражениями; первичные и внешние ключи. Композиция (соединение) отношений по ключу.

DB3 Проектирование БД как декомпозиция отношений. Нормальные формы БД.

DB4 Модификация БД. Нарушения целостности БД уровня поля и записи. Ссылочная целостность.

DB5 Эволюция сетевых БД - хост-терминал, файл-сервер, клиент-сервер и 3-уровневая архитектура. Определение БД.

DB6 Декларативный язык SQL как стандарт коммуникации в сетевых БД. Структура SQL. Подъязыка определения структуры данных DDL - создание таблиц и индексов.

DB7 Подъязык модификации данных DML. Команды insert, delete и update.

DB8 Команда Select. Обязательные параметры и опция where. Базовые предикаты SQL.

DB9 Выборка из декартовых произведений и степеней. Разрешение коллизии имен - квалифицированные (полные) имена и алиасы. Опция Join - варианты соединения таблиц.

DB10 Группировка и групповые вычисления. Опции Group By, Having и агрегатные функции.

DB11 Опции Order By, All/Distinct. Объединение запросов - опция union.

DB12 Вложенные подзапросы. Предикаты с выборкой. Соотнесенные и несоотнесенные подзапросы.

DB13 Представления как интерфейс данных. Создание представлений БД как обращение декомпозиции. Проблемы неоднозначности модификации представлений. Представления в архитектуре "клиент-сервер".

DB14 Транзакции как расширение семантики процедур. Команды поддержки транзакций. Понятия буферизации и блокирования. Примеры транзакций.

Информатика, 3. 2
Введение в ООП.

OP1 ООП как оперирование типами. Семантика классов как именованных типов. Синтаксис определения классов. Поля и методы. Конструкторы и деструкторы.

OP2 Инкапсуляция как относительная локализация. Соккрытие от пользователя - опции private и public. Данные как функции доступа, свойства как представления (интерфейс) данных.

OP3 Наследование реализации как уточнение семантики типа. Пополнение и переопределение методов. Соккрытие реализации от другого проектировщика классов - опция protected.

OP4 Полиморфизм как уточнение семантики типа переменной. Динамические методы как поддержка полиморфизма - опции static, dynamic/virtual

OP5 Проблема множественности иерархий. "Симметричное" решение - агрегаты (декартовы произведения классов). "Ассимметричное" решение - именованные интерфейсы. Наследование интерфейса (компонентный подход).

OP6 Системы со сложным поведением - проблема описания взаимодействия многих объектов/субъектов в современном программировании. Автоматные модели. Потoki как генераторы и приемники сообщений. "Тривиальное" поведение.

OP7 Нетривиальное - изменчивое поведение. Автоматы Мили как обработчики сообщений. Самоопределение через локальное состояние (обратная связь).

OP8 "Интеллектуальное" поведение как самоопределение через глобальное состояние внешней среды. Ленты и машины Тьюринга. Расширенный тезис Черча-Тьюринга.

OP9 События как предикаты. Блок-схемы как семантические автоматы. 3 нормальные формы алгоритмов. Событийный стиль в процедурном программировании - управление данными (на примере).

OP10 Событийное программирование в ООП. Программы как тип данных и интерфейс пользователя. События внешние и внутренние. Варианты делегирования обработки сообщений.

OP11 События как исключения. Обработка исключений в Object Pascal.

Задачи экзамена 3 семестра.

СУБД.

Задачи по СУБД - 2 типов.

В задачах первого типа предлагается написать команду языка SQL, решающую некоторую поставленную на естественном языке задачу. Неизбежная при этом неоднозначность, как обычно трактуется в пользу экзаменуемого. Скажем, "Выбрать информацию о продавцах?" может означать выборку значения ключа, всех (*) или некоторой совокупности полей одной таблицы Salesmen, полной информации о продавцах, включающей информацию о покупателях данного продавца, сделанных ими покупках и товарах и проч. Неявно предполагается синтаксически наиболее простой, т.е. - первый или второй из перечисленных вариантов.

В задачах второго типа предлагается реализовать заданную команду SQL как обработку файлов в терминах процедурного программирования (т.е. в Паскале). Пожалуй, единственная возникающая при этом существенная проблема - эффективная, однопроходная реализация предполагает наличие определенного порядка компонент, но Паскаль не поддерживает понятия логического (динамического, изменяемого) порядка (т.е. индексов). Потому - при решении предполагаем, что компоненты расположены "так, как мне нужно, как того требует решение задачи" - т.е. то, нужный порядок реализован физически. Определение этого порядка выносится в комментарий - равно как и словесная постановка задачи.

Все задачи ссылаются на следующее определение таблиц и связей модельной БД "Заказы" - см. диаграмму. По сравнению с определением в лекциях, для упрощения решений добавлена лишь связь "одни ко многим" Employee.Id=Customer.Emp_Id. Содержательно - каждый покупатель "прикреплен" к единственному обслуживающему его продавцу, что естественно - скажем, для оптовых фирм.

Задачи первого типа.

DB 1.1 Составить упорядоченный по именам и датам рождения список молодых покупателей. Здесь "молодой" \approx родился позднее 1980 г.

DB 1.2

Выбрать всю информацию о парах "продавец - его покупатель", живущих в разных городах.

DB 1.3 Выбрать всю информацию о парах "продавец - не его покупатель", живущих в одном городе.

DB 1.4 Выбрать продавцов, все покупатели которых живут в одном городе.

DB 1.5 Выбрать продавцов, все покупатели которых живут в том же самом городе (что и продавец).

DB 1.6 Выбрать продавцов, у которых существуют покупатели, живущие в том же самом городе (что и продавец).

DB 1.7 Выбрать продавцов, у которых есть по меньшей мере десятков покупателей, живущие в том же самом городе (что и продавец).

DB 1.8 Выбрать продавцов, обслуживших (за все время) более 100 покупателей.

DB 1.9 Выбрать продавцов, обслуживших за текущий месяц более 100 покупателей.

DB 1.10 Выбрать продавцов, которые моложе (в среднем) всех покупателей.

DB 1.11 Выбрать продавцов, которые моложе каждого из покупателей.

DB 1.12 Выбрать продавцов, которые моложе по меньшей мере одного из своих покупателей.

DB 1.13 Выбрать продавцов, которые моложе (в среднем) всех своих покупателей.

DB 1.14 Выбрать продавцов, которые моложе всех покупателей, живущих в том же городе.

DB 1.15 Выбрать пары продавцов-земляков. Дубликаты в выборку не включать.

DB 1.16 Составить список "фамилия-возраст" казанских покупателей и продавцов.

DB 1.17 Вычислить средний возраст совершеннолетних покупателей, по городам.

DB 1.18 Составить список "город - средний возраст покупателей по городу". Результаты вне интервала (18-55) не включать.

DB 1.19 Увеличить вдвое комиссионные продавцам, все покупатели которых живут вне Казани.

DB 1.20 Удалить информацию о покупателях, не сделавших за текущий год ни одной покупки.

Задачи второго типа.

DB 2.1

Select name, birthday from Customers where year(birthday)>1980 order by 1,2

DB 2.2

Select Employee.id, Customer.id from Employee Inner Join Customer on Employee.id=Customer.emp_ref where Employee.city=Customer.city

DB 2.3

Select Employee.id, Customer.id from Employee Inner Join Customer on Employee.id<>Customer.emp_ref where Employee.city=Customer.city

DB 2.4

Select * from Employee where 1=(Select count(*) from Customer where emp_id=Employee.Id)

DB 2.5

Select * from Employee where city= All (Select city from Customer where emp_id=Employee.Id)

DB 2.6

Select * from Employee where not Exists (Select city from Customer where emp_id=Employee.Id and Customer.city=Employee.city)

DB 2.7

Select * from Employee where 10<=(Select count(*) from Customer where emp_id=Employee.Id and Customer.city=Employee.city)

DB 2.8

Select * from Employee where 100<=(Select count(*) from Customer where emp_id=Employee.Id)

DB 2.9

Select * from Employee where 1000<(Select Sum(Amount*Price) from Orders Join Product on Orders.prod_ref=Product_Id where Orders.Emp_ref=Employee.id)

DB 2.10

Select * from Employee where birthday< (Select Avg(birthday) from Customers)

DB 2.11

Select * from Employee where birthday< (Select Min(birthday) from Customers)

DB 2.12

Select * from Employee where exists (Select id from Customers where Customer.emp_ref=Employee.id and Customer.birthday>Employee.birthday)

DB 2.13

Select * from Employee where birthday< (Select Avg(birthday) from Customers where Customer.emp_ref=Employee.id)

DB 2.14

Select * from Employee where birthday< All (Select birthday from Customers where Customer.city=Employee.city)

DB 2.15

Select * from Employee E1, Employee E2 where E1.city=E2.city and E1.id<E2.id

DB 2.16

Select name, Year(date())-Year(birthday) from Employee where city='Казань' union Select name, Year(date())-Year(birthday) from Customer where city='Казань'

DB 2.17

Select AVG(Year(date())-Year(birthday)), city from Customer group by city where Year(date())-Year(birthday)>=18

DB 2.18

Select AVG(Year(date())-Year(birthday)), city from Customer group by city having AVG(Year(date())-Year(birthday)) between (18,55)

DB 2.19

Update Employee set Comm=Comm*2 where not exists (select id from Customer where Customer.emp_ref=Employee.id and city='Казань')

DB 2.20

Delete from Customer where not exists (Select * from Orders where orders.cust_ref=customer.id and year(start)=year(date()))

Задачи экзамена 3 семестра.

ООП.

На деле, все предлагаемые задачи - задачи экзамена 2 семестра. Предлагается лишь оформить их в стиле объектного программирования. Последнее, в каноническом понимании, как известно, предполагает описание

- 1) Строения - в терминах инкапсуляция-наследование-полиморфизм
- 2) Поведения - в терминах обработки сообщений

Содержательно, знание и, по необходимости и возможности, объяснение предложенных решений в терминах соответствующих понятий - полиморфизм, обработка сообщений и интерфейс - не только приветствуется, но и неявно предполагается.

Фактически/формально же, для решения задач предлагается использование лишь языковых средств описания инкапсуляции и наследования. Точнее, все без исключения используемые структуры данных - как абстрактные, так и определенные в Паскале производные типы - должны быть определены как наследники описываемых ниже классов.

Эти классы считаются предопределенными в некотором модуле Structures, который, в свою очередь, ссылается на некоторые определения типов, находящиеся в модуле Base:

tInfo - содержимое вершины,

tPosition - (абсолютная) позиция вершины,

tDirection - (относительное) направление от вершины

Предполагается, что программист имеет доступ к модулю Base и может менять не только определение, но (если нужно) и название типа tInfo - скажем, tListInfo, tStackInfo, tNodeInfo для именованного содержимого компонента списка, стека, дерева и т.д. Подобные замены оформляются в виде комментария.

Автоматы

Определяемые ниже замкнутые автоматы (т.е. автоматы без входа/выхода) фактически в задачах не используются, но нужны для понимания определений, следующих далее.

Класс tAutomata описывает автоматы - стрелочные диаграммы, вершины которых могут находиться в позициях tPosition и содержать значения типа tInfo, а стрелки помечены значениями типа tDirection. Одна из вершин - назовем ее root - явно выделяется как начальная, исходная, "голова" или "корень".

Неявно, предполагается стандартная ссылочная реализация, но для использования существенно лишь то, что класс предлагает следующий набор возможностей (интерфейс, сервис):

- 1) constructor Create(RootInfo:tInfo); - создание автомата из единственной начальной вершины (голова, корня).
- 2) property Info:tInfo read GetInfo write SetInfo; - содержимое текущей вершины (чтение/запись)
- 3) property Position:tPosition read GetPosition - позиция текущей вершины
- 4) property Root:tPosition read GetRoot - позиция корневой вершины (Position и Root - свойства "только для чтения" - нет никакой необходимости изменять позиции вершин - особенно, корня!)
- 5) function CanMove(Direction:tDirection):boolean; проверка возможности перехода в заданном направлении
- 6) procedure Move(Direction:tDirection); переход/сдвиг текущей вершины в заданном направлении; неопределена, если CanMove=false
- 7) procedure Add(Direction:tDirection);

8) procedure Delete(Direction:tDirection); - добавление и удаление пустой вершины в заданном направлении после текущей (и - перед первой вершиной, следующей в заданном направлении, если таковая есть).

9) procedure GoTo(Position:tPosition) - переход/скачок на указанную позицию; соответствующая вершина становится текущей. Неопределена, если вершины с такой позицией не существует. Заметим - вершина root существует всегда и не может быть удалена по определению. Не путать GoTo с относительным движением - Move!

Записи

(трактовка - именованные декартовы произведения автоматов) естественно реализуются как классы без собственных методов - но, конечно, с методами класса tObject и операцией выборки.

Списки

Здесь и далее в задачах под списком (компонент типа tInfo) понимается тип tList, определяемый в точности как tAutomata, при следующем определении типа направлений, с естественной семантикой

tDirection={next}; {можно сдвигаться только вперед, на следующий}

Итак, все остальные линейные структурные типы - массивы, файлы, стеки, очереди и т.п. - реализуются как наследники tList.

Деревья

Класс tTree описывает бинарные деревья и предлагает тот же интерфейс, что и автоматы, при следующем определении типа направления

tDirection={left,right}; {можно сдвигаться вниз налево и направо}

Задачи блока 3.

(Если не хотим мороки с определением значений) пишем процедуры, не программы!

Списки

List 1. Найти длину l самого длинного слова в тексте t и само это слово v. Текст и слова заданы списком символов.

List 2. Удалить из текста t все вхождения заданного слово w. Текст и слова заданы списком символов.

List 3. Вставить заданное слово v1 после первого вхождения заданного слова v2 в данном тексте t. Текст и слова заданы списком символов.

List 4. Заменить первое вхождение в текст t заданного слова v1 на слово v2. Текст и слова заданы списком символов.

List 5. Проверить вхождение слова в текст (как сплошного фрагмента). Текст и слова заданы списком символов.

List 6. Включить элементы заданного списка в упорядоченный список, с сохранением упорядоченности

List 7. Вставить в список заданное значение x перед каждым значением y

List 8. Сортировка списка включением.

List 9. Породить список из положительных компонент данного списка.

Упорядоченные списки и сортировка.

List 10. Проверка упорядоченности числового списка.

List 11. Найти пересечение $I_3 = I_1 \cap I_2$ двух упорядоченных линейных списков чисел I_1, I_2 .

List 12. Найти объединение $I_3 = I_1 \cup I_2$ двух упорядоченных линейных списков I_1, I_2 .

List 13. Найти разность $I_3 = I_1 - I_2$ двух упорядоченных линейных списков I_1, I_2 .

List 14. Обращение списка, преобразование: список \diamond список , с изменением порядка на обратный

List 15. Дан список точек плоскости (запись/полярные координаты). Выяснить, лежат ли они на заданной прямой (заданы коэффициенты линейного уравнения)

List 16. Дан список точек плоскости (запись/декартовы координаты). Выяснить, лежат ли они на окружности заданного радиуса с центром в начале координат.

Деревья.

Tree 1. Найти сумму компонент двоичного дерева над типом real.

Tree 2. Вычислить минимальную компоненту двоичного дерева над типом real.

Tree 3. Выяснить, обладает ли данное дерево свойством - сумма каждого уровня больше суммы предыдущего.

Tree 4. Выяснить, обладает ли данное дерево свойством - компонента каждого уровня больше всех компонент предыдущего.

Tree 5. Выяснить, является ли данное дерево деревом поиска.

Tree 6. Выяснить, обладает ли данное дерево свойством - каждая дочерняя компонента больше по значению родительской.

Tree 7. Декодировать текст, записанный азбукой морзе. Кодировка задана бинарным деревом букв.

Tree 8. Закодировать текст азбукой морзе. Кодировка задана бинарным деревом букв.

Tree 9. Подсчитать количество вхождений каждого слова в заданный текстовый файл.
Результат - дерево количеств.

Tree 10. Преобразовать дерево в список

Tree 11. Преобразовать список в дерево поиска

7.1. Основная литература:

1. Информатика: Курс лекций. Учебное пособие / Е.Л. Федотова, А.А. Федотов. - М.: ИД ФОРУМ: ИНФРА-М, 2011. - 480 с.

<http://znanium.com/catalog.php?bookinfo=204273>

2. Информатика. Базовый курс : учебное пособие для студентов высших технических учебных заведений / под ред. С. В. Симоновича .? 2-е изд. ? Санкт-Петербург [и др.] : Питер, 2008 .? 639 с. :

3. Андрианова, А.А. Практикум по курсу "Алгоритмизация и программирование". Часть 1 / А.А. Андрианова, Т.М. Мухтарова. - Казань: Казанский государственный университет, 2008. - 96 с.

4. Андрианова А.А., Мухтарова Т.М. Практикум по курсу "Алгоритмизация и программирование" - часть 1. - Казанский государственный университет, 2008.

http://libweb.ksu.ru/ebooks/09_63.pdf

5. Практикум работы на ЭВМ: учебное пособие / [В. С. Кугураков и др.]; Казан. гос. ун - т, Фак. вычисл. математики и кибернетики. - Казань: КГУ, 2008. - 111

6. Могилев, А. В. Методы программирования. Компьютерные вычисления / А. В. Могилев, Л. В. Листрова. ? СПб.: БХВ-Петербург, 2008. ? 320 с. URL:

<http://znanium.com/bookread.php?book=350418>

7.2. Дополнительная литература:

1. Новиков, Ф. А. Дискретная математика для программистов: Учеб. пособие / Ф.А.Новиков. - 2 - е изд. - СПб. и др.: Питер, 2004. - 363 с.

2. Delphi. Программирование на языке высокого уровня : Учеб. для вузов, обучающихся по направлению подгот. дипломир. спец. "Информатика и вычислительная техника" / В.В.Фаронов .? СПб. и др. : Питер, 2004 .? 639с. : ил. ? (Учебник для вузов) .? Библиогр.: с.628 .? Алф. указ.: с.629-639 .? ISBN 5-8046-0008-7.

3. Информатика и программирование : учебник для студ. вузов / Е. П. Истомин, С. Ю. Неклюдов, В. И. Романченко .? СПб. : Андреевский изд. дом, 2006 .? 248 с. ? Библиогр.: с.243-247 .? ISBN 5-902894-05-0 : р.256.00.

4. Структуры данных и алгоритмы : Учеб. пособие / Альфред В.Ахо, Джон Э.Хопкрофт, Джеффри Д.Ульман ; Пер. с англ. и ред. А.А.Минько .? М. и др. : Издат. дом "Вильямс", 2000 .? 382с. : ил. ? Библиогр.: с.369-374 .? Предм. указ.: с.375-382 .? ISBN 5-8459-0122-7 (рус.) : 198.90 .?

7.3. Интернет-ресурсы:

О.Кузнецов. Алгоритмы и теория вычислений - <http://www.intuit.ru/studies/courses/555/411/info>

В.Иванников. Введение в алгоритмы. - <http://www.intuit.ru/studies/courses/1010/320/info>

Д.Швед. Алгоритмы: построение и анализ - <http://www.intuit.ru/studies/courses/534/390/info>

Информатика. Базовый курс: учебное пособие для студентов высших технических учебных заведений / под ред. С. В. Симоновича. 2-е изд.. Санкт-Петербург [и др.]: Питер, 2007. 639 с.
URL: http://z3950.ksu.ru/bcover/0000758670_con.pdf - URL:
http://z3950.ksu.ru/bcover/0000758670_con.pdf

М.Фуругян. Алгоритмы и модели вычислений - <http://www.intuit.ru/studies/courses/533/389/info>

8. Материально-техническое обеспечение дисциплины(модуля)

Освоение дисциплины "Дополнительные главы информатики" предполагает использование следующего материально-технического обеспечения:

Лекционные занятия по дисциплине проводятся в аудитории, оснащенной доской и мелом(маркером).

Программа составлена в соответствии с требованиями ФГОС ВПО и учебным планом по направлению 010400.62 "Прикладная математика и информатика" и профилю подготовки Математическая кибернетика .

Автор(ы):

Ахтямов Р.Б. _____

Бухараев Н.Р. _____

Самитов Р.К. _____

"__" _____ 201__ г.

Рецензент(ы):

Еникеев А.И. _____

"__" _____ 201__ г.