

ТАШКЕНТСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ им.А.Р. БЕРУНИ

ФАКУЛЬТЕТ ЭЛЕКТРОНИКА И АВТОМАТИКА

КАФЕДРА «ИНФОРМАТИКА»

КУРСОВАЯ РАБОТА

По предмету: «Информатика и информационные технологии»

Выполнил: студент гр.108-13ЭА Мажидов М.

Приняла: Каримова Д.

Ташкент 2014

Содержание

Введение.....	3
Постановка задачи.....	14
Основная часть.....	15
Блок схема.....	24
Программа.....	25
Обоснование программы.....	28
Заключение.....	30
Литература.....	31

Введение

Современное материальное производство и другие сферы деятельности все больше нуждаются в информационном обслуживании, переработке огромного количества информации. Универсальным техническим средством обработки любой информации является компьютер, который играет роль усилителя интеллектуальных возможностей человека и общества в целом, а коммуникационные средства, использующие компьютеры, служат для связи и передачи информации. Появление и развитие компьютеров - это необходимая составляющая процесса информатизации общества.

Информатизация общества является одной из закономерностей современного социального прогресса. Этот термин все настойчивее вытесняет широко используемый до недавнего времени термин «компьютеризация общества». При внешней схожести этих понятий они имеют существенное различие.

При *компьютеризации общества* основное внимание уделяется развитию и внедрению технической базы компьютеров, обеспечивающих оперативное получение результатов переработки информации и ее накопление.

При *информатизации общества* основное внимание уделяется комплексу мер, направленных на обеспечение полного использования достоверного, исчерпывающего и своевременного знания во всех видах человеческой деятельности.

Таким образом, «информатизация общества» является более широким понятием, чем «компьютеризация общества», и направлена на скорейшее овладение информацией для удовлетворения своих потребностей. В понятии «информатизация общества» акцент надо делать не столько на технических средствах, сколько на сущности и цели социально-технического прогресса.

Компьютеры являются базовой технической составляющей процесса информатизации общества.

Информатизация на базе внедрения компьютерных и телекоммуникационных технологий является реакцией общества на потребность в существенном увеличении производительности труда в информационном секторе общественного производства, где сосредоточено более половины трудоспособного населения. Так, например, в информационной сфере США занято более 60% трудоспособного населения, в СНГ — около 40%.

С современной точки зрения использование телефона в первые годы его существования выглядит довольно смешно. Руководитель диктовал сообщение своему секретарю, который затем отправлял его из телефонной комнаты. Телефонный звонок принимали в аналогичной комнате другой компании, текст фиксировали на бумаге и доставляли адресату. Потребовалось много времени, прежде чем телефон стал таким распространенным и привычным способом сообщения, чтобы его стали использовать, так, как мы это делаем сегодня: сами звоним в нужное место, а с появлением сотовых телефонов – и конкретному человеку.

В наши дни компьютеры, в основном, применяются как средства создания и анализа информации, которую затем переносят на привычные носители (например, бумагу). Но теперь, благодаря широкому распространению компьютеров и созданию Интернета, впервые можно при помощи своего компьютера общаться с другими людьми через их компьютеры. Необходимость использования распечатанных данных для передачи коллегам устраняется подобно тому, как бумага исчезла из телефонных переговоров. Сегодняшний день, благодаря использованию Web, можно сравнить с тем временем, когда люди перестали записывать текст телефонных сообщений: компьютеры (и их связь между собой посредством Интернета) уже настолько широко распространены и привычны, что мы начинаем использовать их принципиально новыми

способами. WWW – это начало пути, на котором компьютеры по – настоящему станут средствами связи.

Интернет предоставляет беспрецедентный способ получения информации. Каждый, имеющий доступ к WWW, может получить всю имеющуюся на нем информацию, а также мощные средства ее поиска. Возможности для образования, бизнеса и роста взаимопонимания между людьми становятся просто ошеломляющими. Более того, технология Web позволяет распространять информацию повсюду. Простота этого способа не имеет аналогов в истории. Для того чтобы сделать свои взгляды, товары или услуги известными другим, больше нет необходимости покупать пространство в газете или журнале, платить за время на телевидении и радио. Web делает правила игры одинаковыми для правительства и отдельных лиц, для малых и больших фирм, для производителей и потребителей, для благотворительных и политических организаций. WorldWideWeb (WWW) на Интернетe – это самый демократичный носитель информации: с его помощью любой может сказать и услышать сказанное без промежуточной интерпретации, искажения и цензуры, руководствуясь определенными рамками приличия. Интернет обеспечивает уникальную свободу самовыражения личности и информации.

Подобно использованию внутренних телефонов компаний для связи сотрудников между собой и внешним миром, Web применяется как для связи внутри организации, так и между организациями и их потребителями, клиентами и партнерами. Та же самая технология Web, которая дает возможность небольшим фирмам заявить о себе на Интернетe, крупной компанией может использоваться для передачи данных о текущем состоянии проекта по внутренней интрасети, что позволит ее сотрудникам всегда быть более осведомленными и, значит, более оперативным по сравнению с небольшими, проворными конкурентами. Применение интрасети внутри организации для того, чтобы сделать информацию более доступной для своих членов, также является шагом вперед по сравнению с прошлым.

Теперь, вместо того, чтобы хранить документы в запутанном компьютерном архиве, появилась возможность (под контролем средств защиты) легко производить поиск и описание документов, делать ссылки на них и составлять указатели. Благодаря технологии Web бизнес, равно как и управления, становится более эффективным.

1.1. Информационные технологии обработки данных.

Информационная технология обработки данных предназначена для решения хорошо структурированных задач, по которым имеются необходимые входные данные и известны алгоритмы и другие стандартные процедуры их обработки. Эта технология применяется на уровне операционной (исполнительской) деятельности персонала невысокой квалификации в целях автоматизации некоторых рутинных постоянно повторяющихся операций управленческого труда. Поэтому внедрение информационных технологий и систем на этом уровне существенно повысит производительность труда персонала, освободит его от рутинных операций, возможно, даже приведет к необходимости сокращения численности работников.

На уровне операционной деятельности решаются следующие задачи:

- обработка данных об операциях, производимых фирмой;
- создание периодических контрольных отчетов о состоянии дел в фирме;
- получение ответов на всевозможные текущие запросы и оформление их в виде бумажных документов или отчетов.

Примером может послужить ежедневный отчет о поступлениях и выдачах наличных средств банком, формируемый в целях контроля баланса наличных средств, или же запрос к базе данных по кадрам, который позволит получить данные о требованиях, предъявляемых к кандидатам на занятие определенной должности.

Существует несколько особенностей, связанных с обработкой данных, отличающих данную технологию от всех прочих:

- выполнение необходимых фирме задач по обработке данных. Каждой фирме предписано законом иметь и хранить данные о своей деятельности, которые можно использовать как средство обеспечения и поддержания контроля на фирме. Поэтому в любой фирме обязательно должна быть информационная система обработки данных и разработана соответствующая информационная технология;
- решение только хорошо структурированных задач, для которых можно разработать алгоритм;
- выполнение стандартных процедур обработки. Существующие стандарты определяют типовые процедуры обработки данных и предписывают их соблюдение организациями всех видов;
- выполнение основного объема работ в автоматическом режиме с минимальным участием человека;
- использование детализированных данных. Записи о деятельности фирмы имеют детальный (подробный) характер, допускающий проведение ревизий. В процессе ревизии деятельность фирмы проверяется хронологически от начала периода к его концу и от конца к началу;
- акцент на хронологию событий;
- требование минимальной помощи в решении проблем со стороны специалистов других уровней.

Хранение данных: многие данные на уровне операционной деятельности необходимо сохранять для последующего использования либо здесь же, либо на другом уровне. Для их хранения создаются базы данных.

Создание отчетов (документов): в информационной технологии обработки данных необходимо создавать документы для руководства и работников фирмы, а также для внешних партнеров. При этом документы могут создаваться как по запросу или в связи с проведенной фирмой операцией, так и периодически в конце каждого месяца, квартала или года.

1.1.1. Информационная технология управления.

Целью информационной технологии управления является удовлетворение информационных потребностей всех без исключения сотрудников фирмы, имеющих дело с принятием решений. Она может быть полезна на любом уровне управления.

Эта технология ориентирована на работу в среде информационной системы управления и используется при худшей структурированности решаемых задач, если их сравнивать с задачами, решаемыми с помощью информационной технологии обработки данных.

Информационная технология управления идеально подходит для удовлетворения сходных информационных потребностей работников и различных функциональных подсистем (подразделений) или уровней управления фирмой. Поставляемая ими информация содержит сведения о прошлом, настоящем и вероятном будущем фирмы. Эта информация имеет вид регулярных или специальных управленческих отчетов.

Для принятия решений на уровне управленческого контроля информация должна быть представлена в агрегированном виде, так, чтобы просматривались тенденции изменения данных, причины возникших отклонений и возможные решения. На этом этапе решаются следующие задачи обработки данных:

- оценка планируемого состояния объекта управления;
- оценка отклонений от планируемого состояния;
- выявление причин отклонений;
- анализ возможных решений и действий.

Информационная технология управления направлена на создание различных видов отчетов. Регулярные отчеты создаются в соответствии с

установленным графиком, определяющим время их создания, например месячный анализ продаж компании.

Специальные отчеты создаются по запросам управленцев или когда в компании произошло что-то незапланированное. И те, и другие виды отчетов могут иметь форму суммирующих, сравнительных и чрезвычайных отчетов.

В суммирующих отчетах данные объединены в отдельные группы, отсортированы и представлены в виде промежуточных и окончательных итогов поотдельным полям.

Сравнительные отчеты содержат данные, полученные из различных источников или классифицированные по различным признакам и используемые для целей сравнения.

Чрезвычайные отчеты содержат данные исключительного (чрезвычайного) характера.

Использование отчетов для поддержки управления оказывается особенно эффективным при реализации так называемого управления по отклонениям. Управление по отклонениям предполагает, что главным содержанием получаемых менеджером данных должны являться отклонения состояния хозяйственной деятельности фирмы от некоторых установленных стандартов (например, от ее запланированного состояния). При использовании на фирме принципов управления по отклонениям к создаваемым отчетам предъявляются следующие требования:

- отчет должен создаваться только тогда, когда отклонение произошло;
- сведения в отчете должны быть отсортированы по значению критического для данного отклонения показателя;
- все отклонения желательно показать вместе, чтобы менеджер мог уловить существующую между ними связь;
- в отчете необходимо показать, количественное отклонение от нормы.

Основные компоненты: входная информация поступает из систем операционного уровня. Выходная информация формируется в виде

управленческих отчетов в удобном для принятия решения виде. Содержимое базы данных при помощи соответствующего программного обеспечения преобразуется в периодические и специальные отчеты, поступающие к специалистам, участвующим в принятии решений в организации. База данных, используемая для получения указанной информации, должна состоять из двух элементов:

- 1) данных, накапливаемых на основе оценки операций, проводимых фирмой;
- 2) планов, стандартов, бюджетов и других нормативных документов, определяющих планируемое состояние объекта управления (подразделения фирмы).

1.1.2. Информационная технология поддержки принятия решений.

Эффективность и гибкость информационной технологии во многом зависят от характеристик интерфейса, системы поддержки принятия решений. Интерфейс определяет: язык пользователя; язык сообщений компьютера, организующий диалог на экране дисплея; знания пользователя.

Язык пользователя — это те действия, которые пользователь производит в отношении системы путем использования возможностей клавиатуры, электронных карандашей, пишущих на экране, джойстика, «мыши», команд, подаваемых голосом и т.п. Наиболее простой формой языка пользователя является создание форм входных и выходных документов. Получив входную форму (документ), пользователь заполняет его необходимыми данными и вводит в компьютер. Система поддержки принятия решений производит необходимый анализ и выдает результаты в виде выходного документа установленной формы.

Язык сообщений — это то, что пользователь видит на экране дисплея (символы, графика, цвет), данные, полученные на принтере, звуковые выходные сигналы и т.п. Важным измерителем эффективности

используемого интерфейса является выбранная форма диалога между пользователем и системой. В настоящее время наиболее распространены следующие формы диалога: запросно-ответный режим, командный режим, режим меню, режим заполнения пропусков в выражениях, предлагаемых компьютером. Каждая форма в зависимости от типа задачи, особенностей пользователя и принимаемого решения может иметь свои достоинства и недостатки. Долгое время единственной реализацией языка сообщений был отпечатанный или выведенный на экран дисплея отчет или сообщение. Теперь появилась новая возможность представления выходных данных— машинная графика. Она дает возможность создавать на экране и бумаге цветные графические изображения в трехмерном виде. Использование машинной графики, значительно повышающее наглядность и интерпретируемость выходных данных, становится все более популярным в информационной технологии поддержки принятия решений.

Знания пользователя —это то, что пользователь должен знать, работая с системой. К ним относятся не только план действий, находящийся в голове у пользователя, но и учебники, инструкции, справочные данные, выдаваемые компьютером.

Совершенствование интерфейса, системы поддержки принятия решений, определяется успехами в развитии каждого из трех указанных компонентов. Интерфейс должен обладать следующими возможностями:

- манипулировать различными формами диалога, изменяя их в процессе принятия решения по выбору пользователя;
- передавать данные системе различными способами;
- получать данные от различных устройств системы в различном формате;
- гибко поддерживать (оказывать помощь по запросу, подсказывать) знания пользователя.

1.2.3. Информационная технология экспертных систем.

Наибольший прогресс среди компьютерных информационных систем отмечен в области разработки экспертных систем. Экспертные системы дают возможность менеджеру или специалисту получать консультации экспертов по любым проблемам, о которых этими системами накоплены знания.

Решение специальных задач требует специальных знаний. Однако не каждая компания может себе позволить держать в своем штате экспертов по всем связанным с ее работой проблемам или даже приглашать их каждый раз, когда проблема возникла. Главная идея использования технологии экспертных систем заключается в том, чтобы получить от эксперта его знания и, загрузив их в память компьютера, использовать всякий раз, когда в этом возникнет необходимость. Все это делает возможным использовать технологию экспертных систем в качестве советующих систем.

Сходство информационных технологий, используемых в экспертных системах и системах поддержки принятия решений, состоит в том, что обе они обеспечивают высокий уровень поддержки принятия решений. Однако имеются три существенных различия.

Первое связано с тем, что решение проблемы в рамках систем поддержки принятия решений отражает уровень её понимания пользователем и его возможности получить и осмыслить решение. Технология экспертных систем, наоборот, предлагает пользователю принять решение, превосходящее его возможности.

Второе отличие указанных технологий выражается в способности экспертных систем пояснять свои рассуждения в процессе получения решения. Очень часто эти пояснения оказываются более важными для пользователя, чем само решение.

Третье отличие связано с использованием нового компонента информационной технологии — знаний.

Основными компонентами информационной технологии, используемой в экспертной системе, являются: интерфейс пользователя, база знаний, интерпретатор, модуль создания системы.

Менеджер (специалист) использует интерфейс для ввода информации и команд в экспертную систему и получения выходной информации из нее. Команды включают в себя параметры, направляющие процесс обработки знаний. Информация обычно выдается в форме значений, присваиваемых определенным переменным.

Технология экспертных систем предусматривает возможность получать в качестве выходной информации не только решение, но и необходимые объяснения.

Различают два вида объяснений:

- объяснения, выдаваемые по запросам. Пользователь в любой момент может потребовать от экспертной системы объяснения своих действий;
- объяснения полученного решения проблемы. После получения решения пользователь может потребовать объяснений того, как оно было получено. Система должна пояснить каждый шаг своих рассуждений, ведущих к решению задачи. Хотя технология работы с экспертной системой не является простой, пользовательский интерфейс этих систем является дружелюбным и обычно не вызывает трудностей при ведении диалога.

База знаний содержит факты, описывающие проблемную область, а также логическую взаимосвязь этих фактов. Центральное место в базе знаний принадлежит правилам. Правило определяет, что следует делать в данной конкретной ситуации, и состоит из двух частей: условия, которое может выполняться или нет, и действия, которое следует произвести, если условие выполняется.

Все используемые в экспертной системе правила образуют систему правил, которая даже для сравнительно простой системы может содержать несколько тысяч правил.

Интерпретатор - часть экспертной системы, производящая в определенном порядке обработку знаний (мышление), находящихся в базе знаний. Технология работы интерпретатора сводится к последовательному рассмотрению совокупности правил (правило за правилом). Если условие, содержащееся в правиле, соблюдается, выполняется определенное действие, и пользователю предоставляется вариант решения его проблемы.

Кроме того, во многих экспертных системах вводятся дополнительные блоки: база данных, блок расчета, блок ввода и корректировки данных. Блок расчета необходим в ситуациях, связанных с принятием управленческих решений. При этом важную роль играет база данных, где содержатся плановые, физические, расчетные, отчетные и другие постоянные или оперативные показатели. Блок ввода и корректировки данных используется для оперативного и своевременного отражения текущих изменений в базе данных.

Модуль создания системы - служит для создания набора (иерархии) правил. Существуют два подхода, которые могут быть положены в основу модуля создания системы: использование алгоритмических языков программирования и использование оболочек экспертных систем.

Оболочка экспертных систем представляет собой готовую программную среду, которая может быть приспособлена к решению определенной проблемы путем создания соответствующей базы знаний. В большинстве случаев использование оболочек позволяет создавать экспертные системы быстрее и легче в сравнении с программированием.

Постановка задачи

Дан список группы и результаты участия студентов в олимпиаде по информатике. Необходимо вывести список студентов успешно участвовавших в олимпиаде и занявших первые 4 места. Список оставшихся вывести в порядке убывания результатов олимпиады.

Основная часть

Объектно-ориентированное программирование

Объектно-ориентированное программирование наилучшим образом предоставляет технологию управления элементами любой сложности, создавая условия для многократного использования программных компонентов и объединения данных с методами их обработки.

Суть объектно-ориентированного программирования заключается в использовании концепции “объектов”, то есть, скорее, образов, чем данных.

Руководящая идея этого подхода заключается в стремлении связать данные с обрабатывающими эти данные методами в единое целое - объект. Объекты имеют характеристики и возможности.

Фактически объектно-ориентированное программирование можно рассматривать как модульное программирование нового уровня, когда вместо во многом случайного, механического объединения процедур и данных акцент делается на их смысловую связь.

Объектная модель способна одинаково хорошо описать как элементы управления графического интерфейса (типа кнопок и раскрывающихся списков), так и реальные объекты (велосипед, самолёт, кот и воду). Таким образом, задача объектно-ориентированного программирования состоит в том, чтобы правильно представить эти объекты на языке программирования.

В языке C++ полностью поддерживаются принципы объектно-ориентированного программирования, включая три кита, на которых оно состоит: инкапсуляцию, наследование и полиморфизм.

Инкапсуляция

Совмещение структур данных с функциями (методами), предназначенными для манипулирования этими данными. Инкапсуляция достигается путём введения класса нового механизма структурирования и типизации данных.

Наследование

Создание новых, производных классов, которые наследуют данные и функции от одного или нескольких ранее определённых базовых классов. При этом возможно переопределение или добавление новых данных и методов. В результате создаётся иерархия классов.

Полиморфизм

Присвоение методу единого имени или идентификатора в рамках иерархии классов таким образом, чтобы любой класс в иерархии имел возможность по своему выполнять связанные с этим методом действия.

Одновременно с появлением и детализацией концепции появились и основанные на ней языки программирования. Одним из первых явился алгоритмический язык Modula 2. Язык программирования TurboPascal, разработанный фирмой Borland, начиная с версии 5.5 стал объектно-ориентированным. Но наиболее последовательно воплощение концепция объектно-ориентированного программирования нашла в алгоритмическом языке C++.

Пусть, например, нужно определить для графической системы тип `shape` (фигура). Проблема состоит в том, что мы не различаем общие свойства фигур (например, фигура имеет цвет, ее можно нарисовать и т.д.) и свойства конкретной фигуры (например, окружность - это такая фигура, которая имеет радиус, она изображается с помощью функции, рисующей дуги и т.д.).

Суть объектно-ориентированного программирования в том, что оно позволяет выражать эти различия и использует их. Язык, который имеет

конструкции для выражения и использования подобных различий, поддерживает объектно-ориентированное программирование. Все другие языки не поддерживают его. Здесь основную роль играет механизм наследования, заимствованный из языка Симула.

Те функции, для которых можно определить заявленный интерфейс, но реализация которых (т.е. тело с операторной частью) возможна только для конкретных фигур, отмечены служебным словом `virtual` (виртуальные). В Симуле и С++ виртуальность функции означает: "функция может быть определена позднее в классе, производном от данного".

Определите, какой класс вам необходим; предоставьте полный набор операций для каждого класса; общность классов выразите явно с помощью наследования.

Если общность между классами отсутствует, вполне достаточно абстракции данных. Насколько применимо объектно-ориентированное программирование для данной области приложения определяется степенью общности между разными типами, которая позволяет использовать наследование и виртуальные функции. В некоторых областях, таких, например, как интерактивная графика, есть широкий простор для объектно-ориентированного программирования. В других областях, в которых используются традиционные арифметические типы и вычисления над ними, трудно найти применение для более развитых стилей программирования, чем абстракция данных. Здесь средства, поддерживающие объектно-ориентированное программирование, очевидно, избыточны.

Нахождение общности среди отдельных типов системы представляет собой нетривиальный процесс. Степень такой общности зависит от способа проектирования системы. В процессе проектирования выявление общности классов должно быть постоянной целью. Она достигается двумя способами: либо проектированием специальных классов, используемых как "кирпичи" при построении других, либо поиском похожих классов для выделения их общей части в один базовый класс.

Для представления на C++ множества взаимозависимых классов можно использовать дружественные классы (§§5.4.1).

Еще один способ выражения общности понятий в языке предоставляют шаблоны типа. Шаблонный класс задает целое семейство классов. Например, шаблонный класс список задает классы вида "список объектов T", где T может быть произвольным типом. Таким образом, шаблонный тип указывает, как получается новый тип из заданного в качестве параметра. Самые типичные шаблонные классы - это контейнеры, в частности, списки, массивы и ассоциативные массивы.

Улучшенный C (поддержка процедурного и модульного программирования)

Минимальная поддержка процедурного программирования включает функции, арифметические операции, выбирающие операторы и циклы. Помимо этого должны быть предоставлены операции ввода-вывода. Базовые языковые средства C++ унаследовал от C (включая указатели), а операции ввода-вывода предоставляются библиотекой.

Самая зачаточная концепция модульности реализуется с помощью механизма отдельной трансляции.

Поддержка абстракции данных

Поддержка программирования с абстракцией данных в основном сводится к возможности определить набор операций (функции и операции) над типом. Все обращения к объектам этого типа ограничиваются операциями из заданного набора. Однако, имея такие возможности, программист скоро обнаруживает, что для удобства определения и использования новых типов нужны еще некоторые расширения языка. Хорошим примером такого расширения является перегрузка операций.

Поддержка объектно-ориентированного программирования

Поддержку объектно-ориентированного программирования обеспечивают классы вместе с механизмом наследования, а также механизм вызова функций-членов в зависимости от истинного типа объекта (дело в том, что возможны случаи, когда этот тип неизвестен на стадии трансляции). Особенно важную роль играет механизм вызова функций-членов. Не менее важны средства, поддерживающие абстракцию данных (о них мы говорили ранее). Все доводы в пользу абстракции данных и базирующихся на ней методов, которые позволяют естественно и красиво работать с типами, действуют и для языка, поддерживающего объектно-ориентированное программирование. Успех обоих методов зависит от способа построения типов, от того, насколько они просты, гибки и эффективны. Метод объектно-ориентированного программирования позволяет определять более общие и

гибкие пользовательские типы по сравнению с теми, которые получаются, если использовать только абстракцию данных.

Итак, мы указали, какую минимальную поддержку должен обеспечивать язык программирования для процедурного программирования, для упрятывания данных, абстракции данных и объектно-ориентированного программирования.

В языке C++Builder 6, наряду с другими структурированными типами, особое место занимают типы структура.

Довольно часто, вполне оправданным, является представление некоторых элементов данных в качестве составных частей другой, более крупной логической единицы. Представляется естественным сгруппировать информацию, например, о номере дома, названии улицы и города в единое целое и назвать адресом, а объединенную информацию о дне, месяце и годе рождения назвать датой.

Например:

Ф.И.О.	Год рождения	Группа	Адрес
Иванков Дмитрий	1983	86-02	Ташкент Уйгур 34
Макеева Анна	1984	87-02	Навои Пушкин 4

Насыров Азиз	1982	99-02	Ташкент Ц-13\40\7
Хамидова Замира	1984	89-02	Бухара Фароби 61
Бондарь Алексей	1983	94-02	Самарканд Ц4\6\1

На языке C++ для описания такого типа данных определен тип структура. В отличие от массива, все элементы которого однотипны, структура может содержать элементы разных типов. В языке C++ структура является видом класса и обладает всеми его свойствами, но во многих случаях достаточно использовать структуры так, как они определены в языке C:

```
struct[ имя_типа ]
{тип_1 элемент_1;
тип_2 элемент_2;
тип_n элемент_n;}
[ список_описателей ];
```

Элементы структуры называются полями структуры и могут иметь любой тип, кроме типа этой же структуры, но могут быть указателями на него.

Если отсутствует имя типа, должен быть указан список описателей переменных, указателей или массивов. В этом случае описание структуры служит определением элементов списка:

```
struct
{
char fio[30];
int date, code;
```

```
floatsalary;  
}stuff[100], *ps; /*определение массива структур и указателя на  
структуру */
```

Если список отсутствует, описание структуры определяет новый тип, имя которого можно использовать в дальнейшем наряду со стандартными типами, например:

```
struct Worker  
{ //описание нового типа Worker  
charfio[30];  
int date, code;  
float salary;  
}; //описание заканчивается точкой с запятой  
Workerstuff[100], *ps; // определение массива типа Worker //  
// и указателя на тип Worker
```

Для инициализации структуры значения ее элементов перечисляют в фигурных скобках в порядке их описания:

```
Struct  
{  
charfio[30];  
int date, code;  
floatsalary;  
}worker = {"Страусенко", 31, 215, 3400.55};
```

Для переменных одного и того же структурного типа определена операция присваивания, при этом происходит поэлементное копирование.

Доступ к полям структуры выполняется с помощью операций выбора . (точка) при обращении к полю через имя структуры и -> при обращении через указатель, например:

```
Worker worker, stuff[100], *ps;  
...  
worker.fio = "Страусенко";
```

```
stuff[8].code = 215;
```

```
ps->salary = 0.12;
```

В простейшем случае запись представляет собой перечень имен и типов. Имена полей выступают в программе как отдельные переменные.

Переменная записи, в этом примере состоит из 4-х полей. Для каждого поля приведена переменная и объявлен ее тип. В том случае, если в программе множество переменных этого типа, удобно вводить этот тип в разделе типов, а переменные описывать в разделе **var**, используя объявленный тип записи.

```
const n=20;
```

```
struct stud
```

```
{
```

```
    char fio[15];
```

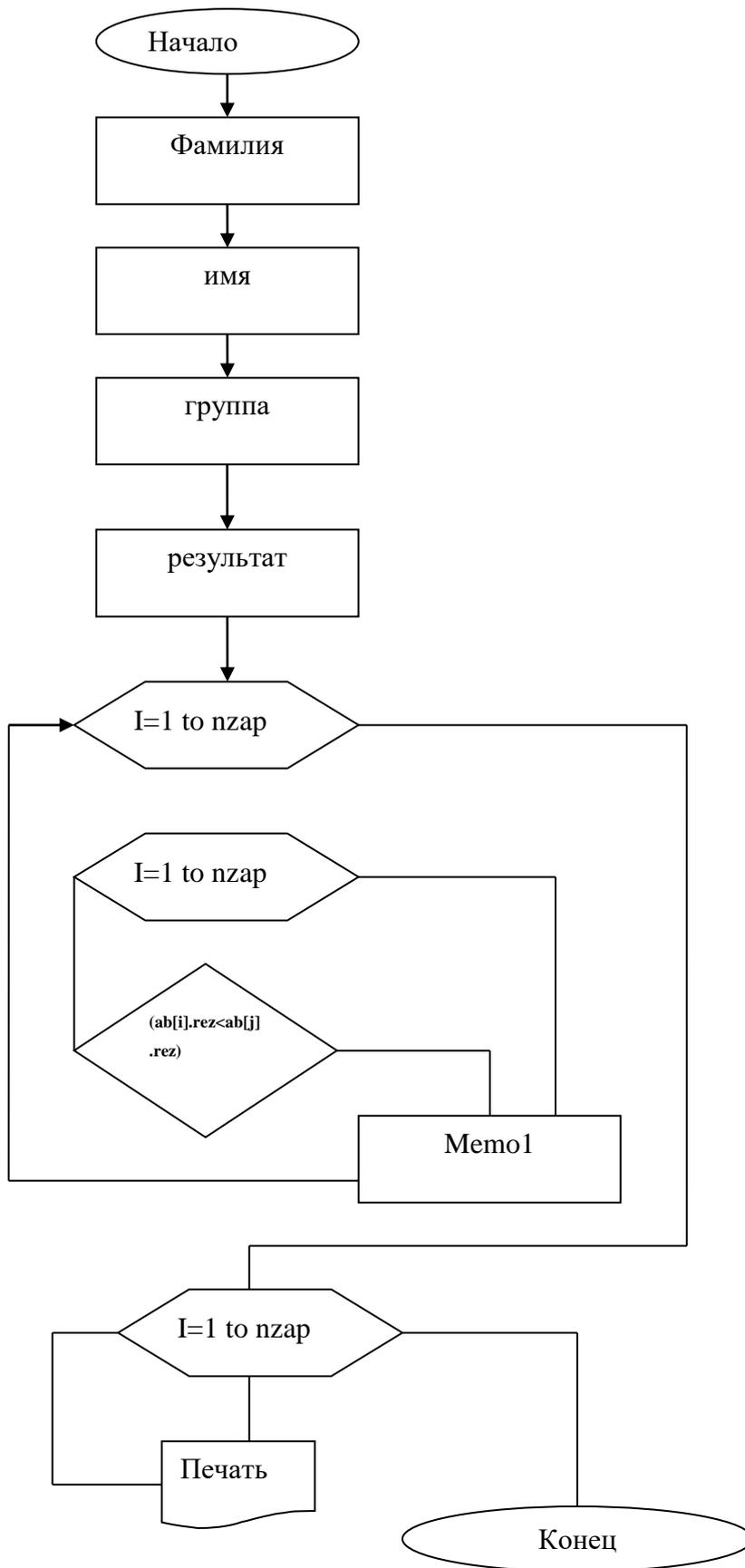
```
    int date;
```

```
    char Pol;;
```

```
    char Adr[15];}
```

При обработке переменных типа структура компилятор языка C++Builder 6 предоставляет возможность прямого доступа к любому полю. Для этого необходимо указать имя переменной и выбранное поле, разделяя их точкой.

Блок схема



Код программы

//-----

```
#include<vcl.h>
```

```
#pragma hdrstop
```

```
#include "Unit1.h"
```

```
//-----
```

```
#pragma package(smart_init)
```

```
#pragma resource "*.dfm"
```

```
TForm1 *Form1;
```

```
typedefstruct
```

```
{ Stringfam;
```

```
    String imya;
```

```
    String grup;
```

```
intrez;} olimpiada;
```

```
olimpiadaab [20]; inti,j,m;    olimpiada k; int n;
```

```
//-----
```

```
__fastcall TForm1::TForm1(TComponent* Owner)
```

```
    : TForm(Owner)
```

```
{
```

```
}
```

```
//-----
```

```
void __fastcall TForm1::BitBtn1Click(TObject *Sender)
```

```

{
    n=n+1;
    ab[n].fam=Edit1->Text;
    ab[n].imya=Edit2->Text;
    ab[n].grup=Edit3->Text;
    ab[n].rez=StrToInt(Edit4->Text);
    Edit1->Text="";
    Edit2->Text="";
    Edit3->Text="";
    Edit4->Text="";
}
//-----
void __fastcall TForm1::BitBtn2Click(TObject *Sender)
{
    Memo2->Clear();
    for (i=1;i<=n-1; i++)
    for (j=i+1;j<=n;j++)
    {
        if (ab[i].rez<ab[j].rez)
            {k=ab[i];
            ab[i]=ab[j];
            ab[j]=k;}}
    for (i=1;i<=n; i++)
    {

```

```
Memo2->Lines->Add(IntToStr(i)+" "+ab[i].fam+"
"+ab[i].imy+" "+ab[i].grup+" "+IntToStr(ab[i].rez));
```

```
}
```

```
}
```

```
//-----
```

```
void __fastcall TForm1::BitBtn3Click(TObject *Sender)
```

```
{
```

```
Close();
```

```
}
```

```
//-----
```

```
void __fastcall TForm1::BitBtn4Click(TObject *Sender)
```

```
{ Memo2->Clear();
```

```
for (i=1;i<=n; i++)
```

```
Memo1->Lines->Add(IntToStr(i)+" "+ab[i].fam+"
```

```
" "+ab[i].imy+" "+ab[i].grup+" "+IntToStr(ab[i].rez));
```

```
}
```

```
//-----
```

Результаты участия студентов в олимпиаде по информатике

Фамилия участника	<input type="text"/>	✓ Записать	✓ Сортировка
Имя участника	<input type="text"/>		
Группа участника	<input type="text"/>	✓ Общий список	⚙ Выход
Результат	<input type="text"/>		

Общий список студентов участвовавших в олимпиаде по информатике

1. Сардоров Сарвар 101-13 111
2. Каримова Салима 102-13 113
3. Фаттоев Фарход 110-13 115
4. Назарова Шохида 117-13 116
5. Идиев Холмурод 118-13 120
6. Жониебеков Жамшид 132-13 117
7. Носиров Одил 13-13 118
8. Тоиров Жасур 14-13 116

Сортировка студентов по убыванию результатов

1. Идиев Холмурод 118-13 120
2. Носиров Одил 13-13 118
3. Жониебеков Жамшид 132-13 117
4. Назарова Шохида 117-13 116
5. Тоиров Жасур 14-13 116
6. Фаттоев Фарход 110-13 115
7. Каримова Салима 102-13 113
8. Сардоров Сарвар 101-13 111

Результаты участия студентов в олимпиаде по информатике

Фамилия участника	Кадыров		
Имя участника	Кадир	✓ Записать	✓ Сортировка
Группа участника	22-13		
Результат	122	✓ Общий список	★ Выход

Общий список студентов участвовавших в олимпиаде по информатике

1. Сардоров Сарвар 101-13 111
2. Каримова Салима 102-13 113
3. Фаттоев Фарход 110-13 115
4. Назарова Шохида 117-13 116
5. Идиев Холмурод 118-13 120
6. Жониебеков Жамшид 132-13 117
7. Носиров Одил 13-13 118
8. Тоиров Жасур 14-13 116

Сортировка студентов по убыванию результатов

1. Идиев Холмурод 118-13 120
2. Носиров Одил 13-13 118
3. Жониебеков Жамшид 132-13 117
4. Назарова Шохида 117-13 116
5. Тоиров Жасур 14-13 116
6. Фаттоев Фарход 110-13 115
7. Каримова Салима 102-13 113
8. Сардоров Сарвар 101-13 111

Заключения

В ходе работы я разработал программу на C++. Создал красивую и удобную программу, который сортирует список группы и результаты участия студентов в олимпиаде по информатике.

Литература

1. Периодические издания (1998 год): *Delphi Informant*, *Delphi Developer*, *Microsoft System Journal*, *Dr. Dobb Journal*, *Компьютерр-Пресс* и др.
2. WWW-серверы: *Borland*, *Miller Friman*, *Turbo Power*, *ProtoView*, *Popkin Software*, *InterSolv*, *AOL* и др.
3. "Delphi Developers Guide", S.Tiexeira & X. Pacheco, SAMS Publishing / Borland PRESS.
4. Каталоги программных продуктов "Delphi Only Tools" ZAC Catalog, "Delphi Power Tools" Informant Communications Group.
5. Джарод Холингвэрт, Дэн Баттерфилд, Боб Сворт, Джэйми Олсон *C++Builder 5. Руководство разработчика*.
6. *Borland C++ Builder 5. Энциклопедия программиста*. Калверт Ч., Рейсдорф К., "ДиаСофт" - 2001, 944 стр.
7. <http://www.codenet.ru/>