Уважаемые студенты групп ДП$_9$-21!

Вашему вниманию представлена лекция на тему «Вступительная лекция. Алгоритм: определение и основные свойства, языки программирования и трансляторы» по дисциплине ОП.05 Основы программирования. Лекция рассчитана на 4 часа (на 1 и 2 пару 18.01)

Задание

1. Прочитать внимательно лекцию.
2. Законспектировать лекцию в рабочую тетрадь не менее 3-4 страницы рукописного текста. В конспекте лекции обязательно должно быть приведены примеры.
3. Решить приведенные в лекции в контрольных вопросах задачи.
4. Дата предоставления полного фотоотчета лекции до 23.01.2023 г.
5. Фотоотчет предоставляется на электронный адрес преподавателя.

С уважением Ганзенко Ирина Владимировна

!!! Если возникнут вопросы обращаться по телефону 0721134803 (вацап), +79591134803 (телеграмм)

disobuch.ganzenko2020@mail.ru

Лекция 1. Вступительная лекция. Алгоритм: определение и основные свойства, языки программирования и трансляторы

План

1. Основные этапы решения задач на компьютере
2. Языки программирования
3. Трансляторы
4. Язык программирования Паскаль
5. Использование среды программирования ТУРБО ПАСКАЛЬ

5.1 Меню программы и функциональные клавиши

5.2 Работа с текстом программы

6 Контрольные вопросы

1 Основные этапы решения задач на компьютере

Процесс решения задач на компьютере - это совместная деятельность человека и ЭВМ. Этот процесс можно представить в виде нескольких последовательных этапов. На долю человека приходятся этапы, связанные с творческой деятельностью — постановкой, алгоритмизацией, программированием задач и анализом результатов, а на долю компьютера — этапы обработки информации в соответствии с разработанным алгоритмом.

Рассмотрим эти этапы на следующем примере: пусть требуется вычислить сумму двух целых чисел и вывести на экран результат.

Первый этап - постановка задачи. На этом этапе участвует человек, хорошо представляющий предметную область задачи. Он должен четко определить цель задачи, дать словесное описание содержания задачи и предложить общий подход к ее решению. Для задачи вычисления суммы двух целых чисел человек, знающий, как складываются числа, может описать задачу следующим образом: ввести два целых числа, сложить их и вывести сумму в качестве результата решения задачи.

Второй этап - математическое или информационное моделирование. Цель этого этапа - создать такую математическую модель решаемой задачи, которая может быть реализована в компьютере. Существует целый ряд задач, где математическая постановка сводится к простому перечислению формул и логических условий. Для вышеописанной задачи данный этап сведется к следующему: введенные в компьютер числа запомним в памяти под именами A и B, затем вычислим значение суммы этих чисел по формуле A+B, и результат запомним в памяти под именем Summa.

Третий этап - алгоритмизация задачи. На основе математического описания необходимо разработать алгоритм решения.

Алгоритмом называется точное предписание, определяющее последовательность действий исполнителя, направленных на решение поставленной задачи. В роли исполнителей алгоритмов могут выступать люди, роботы, компьютеры.

Используются различные способы записи алгоритмов. Широко распространен словесный способ записи: это записи рецептов приготовления различных блюд в кулинарной книге, инструкции по использованию технических устройств, правила правописания и многие другие. Наглядно представляется алгоритм языком блок-схем.

Четвертый этап — программирование. Программой называется план действий, подлежащих выполнению некоторым исполнителем, в качестве которого может выступать компьютер. Составление программы обеспечивает возможность выполнения алгоритма и соответственно поставленной задачи исполнителем-компьютером. Во многих задачах при программировании на алгоритмическом языке часто пользуются заменой блока алгоритма на один или несколько операторов, введением новых блоков, заменой одних блоков

другими.

Пятый этап - ввод программы и исходных данных в ЭВМ. Программа и исходные данные вводятся в ЭВМ с клавиатуры с помощью редактора текстов, и для постоянного хранения осуществляется их запись на гибкий или жесткий магнитный диск.

Шестой этап - тестирование и отладка программы. На этом этапе происходят исполнение алгоритма с помощью ЭВМ, поиск и исключение ошибок. При этом программисту приходится выполнять работу по проверке работы программы, поиску и исключению ошибок, и поэтому для сложных программ этот этап часто требует гораздо больше времени и сил, чем написание первоначального текста программы. *Отладка программы* - сложный и нестандартный процесс. Исходный план отладки заключается в том, чтобы оттестировать программу на контрольных примерах.

Контрольные примеры стремятся выбрать так, чтобы при работе с ними программа прошла все основные пути блок-схемы алгоритма, поскольку на каждом из путей могут быть свои ошибки, а детализация плана зависит от того, как поведет себя программа на этих примерах: на одном она может зациклиться (т. е. бесконечно повторять одно и то же действие); на другом - дать явно неверный или бессмысленный результат и т. д. Сложные программы отлаживают отдельными фрагментами.

Седьмой этап - исполнение отлаженной программы и анализ результатов. На этом этапе программист запускает программу и задает исходные данные, требуемые по условию задачи.

Полученные в результате решения выходные данные анализируются постановщиком задачи, и на основании этого анализа вырабатываются соответствующие решения, рекомендации, выводы. Например, если при решении задачи на компьютере результат сложения двух чисел 2 и 3 будет 4, то следует сделать вывод о том, что надо изменить алгоритм и программу.

Возможно, что по итогам анализа результатов потребуются пересмотр самого подхода к решению задачи и возврат к первому этапу для повторного выполнения всех этапов с учетом приобретенного опыта. Таким образом, в процессе создания программы некоторые этапы будут повторяться до тех пор, пока мы получим алгоритм и программу, удовлетворяющие показанным выше свойствам.

Языки программирования

Чтобы компьютер выполнил решение какой-либо задачи, ему необходимо получить от человека инструкции, как ее решать. Набор таких инструкций для компьютера, направленный на решение конкретной задачи, называется *компьютерной программой.*

Современные компьютеры не настолько совершенны, чтобы понимать программы, записанные на каком-либо употребляемом человеком языке — русском, английском, японском. Команды, предназначенные для ЭВМ,

необходимо записывать в понятной ей форме. С этой целью применяются языки программирования - искусственные языки, алфавит, словарный запас и структура которых удобны человеку и понятны компьютеру.

В самом общем смысле *языком программирования* называется фиксированная система обозначений и правил для описания алгоритмов и структур данных. Все языки программирования делятся на языки низкого, высокого и сверхвысокого уровня.

Язык программирования низкого уровня - это средство записи инструкций компьютеру простыми командами на аппаратном уровне. Такой язык отражает структуру данного класса ЭВМ и поэтому иногда называется машинно-ориентированным языком. Пользуясь системой команд, понятной компьютеру, можно описать алгоритм любой сложности. Запись программы на этом языке представляет собой последовательность нулей и единиц.

Существенной особенностью языков программирования низкого уровня является жесткая ориентация на определенный тип аппаратуры (систему команд процессора). В стремлении приспособить язык программирования низкого уровня к человеку разработан язык символического кодирования (автокод или язык ассемблера), структура команд которого определяется форматами команд и данными машинного языка. Программа на таком языке ближе человеку, потому что операторы этого языка — те же команды, но они имеют мнемонические названия, а в качестве операндов используются не конкретные адреса в оперативной памяти, а их символические имена.

Более многочисленную группу составляют *языки программирования высокого уровня***,** средства которых допускают описание задачи в наглядном, легко воспринимаемом виде. Отличительной особенностью этих языков является их ориентация не на систему команд той или иной ЭВМ, а на систему операторов, характерных для записи определенного класса алгоритмов. К языкам программирования этого типа относятся: Бейсик, Фортран, Алгол, Паскаль, Си. Программа на языках высокого уровня записывается системой обозначений, близкой человеку (например, фиксированным набором слов английского языка, имеющих строго определенное назначение). Программу на языке высокого уровня проще понять и значительно легче отладить.

К *языкам программирования сверхвысокого уровня* можно отнести Алгол, при разработке которого сделана попытка формализовать описание языка, приведшая к появлению абстрактной и конкретной программ. Абстрактная программа создается программистом, конкретная - выводится из первой. Предполагается, что при таком подходе принципиально невозможно породить неверную синтаксически (а в идеале и семантически) конкретную программу. Язык APL относят к языкам сверхвысокого уровня за счет введения сверхмощных операций и операторов. Запись программ на таком языке получается компактной.

Все вышеперечисленные языки - вычислительные. Более молодые - декларативные (непроцедурные) языки, отличительная черта которых -

задание связей и отношений между объектами и величинами и отсутствие определения последовательности выполнения действий (Пролог). Такие языки сыграли важную роль в программировании, так как они дали толчок к разработке специализированных языков искусственного интеллекта и языков представления знаний.

Трансляторы

Так как текст записанной на Паскале программы не понятен компьютеру, то требуется перевести его на машинный язык. Такой перевод программы с языка программирования на язык машинных кодов называется ***трансляцией*** (translation — перевод), а выполняется он специальными программами - *трансляторами***.**

Существует три вида трансляторов: интерпретаторы, компиляторы и ассемблеры.

Интерпретатором называется транслятор, производящий пооператорную (покомандную) обработку и выполнение исходной программы.

Компилятор преобразует (транслирует) всю программу в модуль на машинном языке, после этого программа записывается в память компьютера и лишь потом исполняется.

Ассемблеры переводят программу, записанную на языке ассемблера (автокода), в программу на машинном языке.

Любой транслятор решает следующие основные задачи:

• анализирует транслируемую программу, в частности определяет, содержит ли она синтаксические ошибки;

• генерирует выходную программу (ее часто называют объектной или рабочей) на языке команд ЭВМ (в некоторых случаях транслятор генерирует выходную программу на промежуточном языке, например, на языке ассемблера);

• распределяет память для выходной программы (в простейшем случае это заключается в назначении каждому фрагменту программы, переменным, константам, массивам и другим объектам своих адресов участков памяти).

Язык программирования Паскаль

Алгоритмический язык высокого уровня Паскаль был разработан в конце 60-х годов профессором Н.Виртом. Он был создан специально для обучения программированию. К основным достоинствам языка Паскаль следует отнести гибкость и надежность, простоту и ясность конструкций, возможность удовлетворения требованиям структурного программирования, наличия набора структурированных типов данных: массивов, записей, записей с вариантами, файлов, множеств, возможность построения новых типов данных.

На базе стандартного Паскаля фирма Borland разработала семейство Паскаль-систем, называемых Турбо Паскалем. Турбо Паскаль пользуется широкой популярностью среди массовых пользователей и профессиональных программистов. Это объясняется наличием очень удобной интегрированной среды и тем, что в его основе лежит мощный язык программирования, представляющий собой расширенную версию языка Паскаль.

За последние годы фирма Borland разработала и выпустила на рынок шесть модификаций этой системы. Каждая из них представляет собой усовершенствование предыдущей. Непрерывное совершенствование системы Турбо Паскаля породило в конце концов очень мощную по своим возможностям систему программирования, отвечающую самым взыскательным требованиям. С помощью Турбо Паскаля можно создавать многие программы — от программ, предназначенных для решения простейших вычислительных задач, до сложных современных систем управления базами данных и операционных систем.

И вместе с тем Турбо Паскаль остается простым в изучении, что позволяет начинающему программисту на его основе изучить методы и способы эффективного программирования.

Использование среды программирования ТУРБО ПАСКАЛЬ

Процесс подготовки программы, записанной на алгоритмическом языке, для выполнения на ЭВМ включает в себя следующие этапы:

- ввод текста программы в ЭВМ;
- компиляция - преобразование программы, записанной на алгоритмическом языке, в машинную программу;
- формирование исполняемой программы (такая программа обычно имеет расширение **.exe**);
- отладка программы (поиск и исправление ошибок).

Реализацией алгоритмического языка программирования является компилятор. Компилятор - это специальная программа, преобразующая текст программы на алгоритмическом языке в программу на машинном языке. Компилятор также проверяет правильность записи конструкций языка программирования и выдает диагностическое сообщение, если обнаруживает ошибку.

В современных языках программирования обычно компилятор дополняют набором специальных программ, освобождающих пользователя от рутинной работы при подготовке исполняемой программы. Примером такого комплекса программ является TP.

Турбо Паскаль - это интегрированная программная система для разработки программ на языке Турбо Паскаль в интерактивном режиме.Среда TP ориентирована на взаимодействие с пользователем с помощью системы меню и окон. Для указания тех или иных действий можно также использовать функциональные клавиши (F1 - F10), расположенные в верхнем ряду клавиатуры.

5.1 Меню программы и функциональные клавиши

С помощью функциональных клавиш можно выполнять следующие действия:

[F1] - обращение к справочной службе;

[F2] - запись редактируемого текста в дисковый файл;

[F3] - чтение текста из дискового файла в окно редактора;

[F4] - выполнение программы до строки, на которой располагается курсор;

[F5] -распахнуть активное окно на весь экран (при повторном нажатии на F5 окно возвращается к прежнему размеру);

[F6] - сделать активным следующее окно;

[F7]- выполнить следующую строку программы; если в строке есть обращение к подпрограмме, то войти в подпрограмму и остановиться перед выполнением первого ее оператора;

[F8] - выполнить следующую строку программы; если в строке есть обращение к подпрограмме, то выполнить ее полностью;

[F9]- компилировать программу, создать **EXE**-файл, но не выполнять;

[F10] - войти в главное меню;

[Alt]+[F9]- компилировать программу в активном окне редактора;

[Ctrl]+[F9] - выполнить программу (компилировать программу, находящуюся в редакторе, сформировать исполняемую программу, загрузить ее в оперативную память и выполнить);

[Alt]+[F5] - открыть окно с результатами выполнения программы.

Рассмотренные выше и многие другие действия можно выполнить с помощью меню. Меню - это участок экрана, предназначенный для диалогового выбора работы, которую должна выполнить ЭВМ. Меню фиксирует некоторое текущее состояние диалоговой среды и предлагает несколько альтернативных путей перехода из этого состояния. Содержащиеся в меню альтернативы называют пунктами или элементами меню, командами или опциями. Каждое конкретное меню реализуется в виде небольшого окна с текстом.

Окно - это участок экрана, предназначенный для обмена информацией между программистом и ЭВМ. Среда TP открывает различные окна в процессе выполнения тех или иных работ.

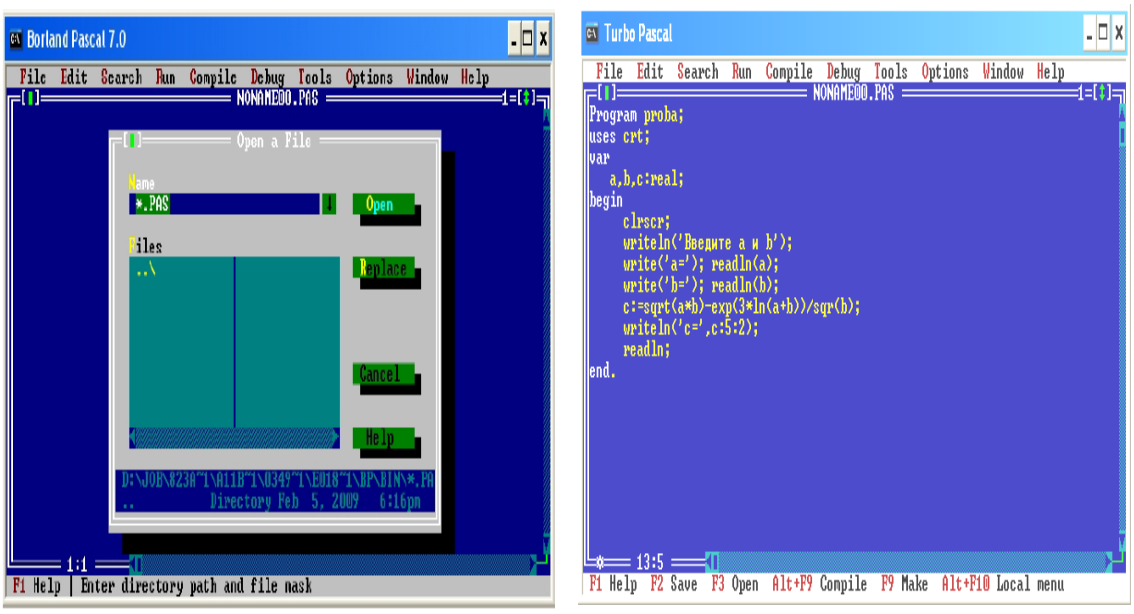

Рисунок 1 – Интерфейс программы Турбо Паскаль

В Турбо Паскале экран разделен на три части (рис.1): вверху - меню возможных наборов команд среды TP; в центре - место для размещения окон; внизу - набор наиболее часто используемых в главном меню функциональных клавиш.

Для активизации команд, соответствующих этим клавишам, можно нажать функциональную клавишу или щелкнуть левой кнопкой мыши по соответствующему месту на экране. Тип окна, расположенного в центральной части экрана, зависит от пункта главного меню. В окнах можно просматривать и редактировать текст программы, просматривать результаты выполнения программы и т. п.

Меню в Турбо Паскале состоит из главного меню и системы подменю. Переход в главное меню из режима редактирования осуществляется нажатием клавиши F10, а возврат в режим редактирования - с помощью клавиши ESC. При активизации пункта главного меню раскрывается соответствующее ему подменю. Активизировать пункт меню можно различными способами:

- щелкнуть левой клавишей мыши по пункту меню;

- нажать клавишу F10 для входа в главное меню, переместить подсветку (с помощью клавиш управления курсором) на соответствующий пункт меню и нажать клавишу Enter;

- нажать клавишу F10, а затем нажать клавишу с буквой, выделенной красным цветом в ключевом слове пункта меню.

Главное меню представляет следующие возможности выбора:

· **File** (файл) - действия с файлами и выход из системы;

· **Edit** (редактировать) - операции с буферной памятью редактора;

· **Search** (искать) - поиск текста, процедуры, функции или места ошибки;

· **Run** (выполнение) - прогон программы;

· **Compile** (компилировать) - компиляция программы;

· **Debug** (отладка) - отладка программы;

· **Tools** (инструментальные средства) – выполнение сервисных функций;

· **Options**(варианты) - задания свойств и режимов работы TP;

· **Windows**(окно) – управление окнами среды TP;

· **Help** (помощь) – обращение к справочной системе.

Каждый из пунктов главного меню имеет подменю, содержащее от трех до десяти различных опций

Меню FILEобеспечивает операции с файлом текста программы: загрузку существующих файлов; создание новых; сохранение редактируемого файла на диске; смены текущей директории и в некоторых других случаях. В этом меню имеются следующие пункты:

· **Open** (открыть) [F3] – открывает новое окно редактора и помещает в него файл с диска. Имя считываемого файла задаётся в дополнительном, так называемом диалоговом, окне, которое открывается при выборе этого режима;

· **New** (новый) - открывает окно редактора для создания нового файла; по умолчанию этому файлу присваивается имя Noname00.pas, которое можно будет изменить при записи файла на диск;

· **Save** (сохранить) [F2] – записывает содержимое активного окна редактора в дисковый файл;

· **Save as** (сохранить как) – записывает содержимое активного окна редактора в дисковый файл под заданным именем;

· **Exit** (выход) [Alt+X]– выход из системы Turbo Pascal.

Меню EDIT. Подменю этого режима используется в основном для удаления, вставки или замены текста:

· **Cut** (вырезать) – удаляет из окна редактора выделенный блок и переносит его во временный буфер;

· **Copy** (копировать) – копирует выделенный блок во временный буфер;

· **Past**e (приклеить) – копирует содержимое временного буфера в окно редактора.

Меню RUN используется для выполнения подготовленных программ в обычном или отладочном вариантах.

· **Run** (счет)-осуществляет полную обработку (прогон) программы, т.е. выполняет компиляцию, компоновку и исполнение программы, подготовленной и находящейся в редакторе.

· **Trace into (трассировка внутри) – осуществляет пошаговое выполнение программы аналогично [F7].**

Меню COMPILE(компиляция*)* осуществляет компиляцию программы или модуля, находящегося в активном окне редактора. В зависимости от того, в каком из подрежимов (Compile, Make, Build) осуществляется

компиляция, обрабатывается либо текст программы из окна редактора, либо текст программы и дополнительные модули, хранящиеся в отдельных файлах.

· **Compile** ([Alt]+ [F9]) - компилирует только ту программу (или модуль), которая загружена в данный момент в память редактора;

· **Make** (создать) [F9] - создает программу: сначала компилируется основной файл, указанный в меню Compile/Primary file или загруженный в редактор, а затем выполняется перекомпиляция тех файлов, составляющих программу, в которых произошли изменения на момент компиляции основного файла программы;

· **Build** (построить*)* –заново транслируются все файлы, составляющие программу, т.е., для всех TPU-файлов отыскивается соответствующий PAS-файл и выполняется его перекомпиляция независимо от того, были сделаны в нем изменения или нет.

Меню **DEBUG (**отладчик*)***–**позволяет управлять процессом отладки программы.

· **Evaluate** (вычислить) – позволяет просмотреть в процессе отладки содержимое любой переменной или найти значение любого выражения с помощью дополнительного окна, содержащего три поля: первое используется для записи имени переменной или выражения, второе – показывает их текущее значение, в третьем можно задать новое значение этой переменной.

· **Output** (вывод)- выводит в окно Output результаты выполнения программы;

· **User screen** (экран пользователя)- вызывает для просмотра выходной экран выполняемой программы.

Меню OPTIONS(параметры) позволяет задавать различные ключи, определяющие режимы работы интегрированной среды, компилятора и компоновщика. Это меню имеет сильно разветвленную структуру, состоящую из подменю нескольких уровней вложенности. Рассмотрим опции:

· **Save options** (сохранить конфигурацию) – позволяет сохранить всю настройку среды (параметры компилятора, компоновщика и самой среды) в специальном файле конфигурации (по умолчанию, имя файла – TURBO.TP);

· Numeric Processing (арифметический сопроцессор) – ключ, позволяющий использовать числовой сопроцессор (режим 8087/80287) либо его не использовать (режим Software). При работе с сопроцессором все операции с вещественными данными реализуются не только для данных типа Real, но и для данных типа Single, Double, Extended и Comp.

5.2 Работа с текстом программы

При входе в TP обычно открывается окно редактора, в котором можно набирать текст новой программы, либо редактировать ранее введенный текст программы (для редактирования текста программы надо открыть меню **File**,

активизировать пункт подменю **Open** и в открывшемся окне ввести или выбрать из списка имя редактируемого файла).

Для перемещения по тексту используются клавиши управления курсором. При нажатии на клавишу с каким-либо символом этот символ размещается на позиции, отмеченной курсором. Для удаления символа на позиции, отмеченной курсором, надо нажать клавишу **Del,**асимвола, расположенного слева от курсора, - клавишу **Backspace**. Для ввода прописных букв используются клавиши **Shift** и **Caps Lock**. Переключение алфавитов осуществляется при помощи комбинации клавиш **Ctrl**+ **Shift**. Для перехода на следующую строку надо нажать клавишу **Enter.**

В редакторе TP есть также средства для работы с выделенным фрагментом текста. Выделенный фрагмент - это символ или любая последовательность символов, выделенных цветом. Такие фрагменты можно копировать, вырезать, вставлять в текст и удалять из текста. При копировании и вырезании выделенный фрагмент помещается в специальную область памяти, называемую буфером, из которого этот фрагмент можно перенести в другое место в тексте. Что находится в буфере в данный момент, можно всегда увидеть, выполнив команды **Edit / Show clipboard**.

Для выделения фрагмента надо:

- установить курсор на первый символ фрагмента;
- нажать клавишу Shift и, не отпуская ее, переместить курсор на последний символ фрагмента. Фрагмент можно выделить также при помощи мыши:
- установить указатель мыши на первый символ фрагмента;
- нажать левую клавишу мыши и, не отпуская ее, переместить указатель мыши на последний символ фрагмента.

Фрагмент можно также выделить, начиная с последнего символа фрагмента.

Для удаления выделенного фрагмента надо выполнить **Edit / Clear**.

Для копирования фрагмента в другое место в тексте необходимо :

- выделить фрагмент текста;
- выполнить **Edit / Copy** (копировать);
- переместить курсор в то место, куда надо вставить фрагмент;
- выполнить команды **Edit / Paste**(вставить);

Для перемещения фрагмента в другое место надо:

- выделить фрагмент;
- выполнить команды **Edit / Cut**(вырезать);
- переместить курсор в то место, куда надо перенести фрагмент;
- выполнить команды **Edit / Paste.**

Отменить предыдущие действия всегда можно с помощью команд **Edit / Undo**.

1.Ввести текст программы в окне редактора.

2.Сохранить текст программы. Для этого открыть меню **File,** выбрать пункт подменю **Save as** и в открывшемся окне ввести имя программы. Этот способ сохранения используют при первом сохранении программы, когда

надо указать имя файла. В дальнейшем, при внесении изменений в текст программы ее также надо сохранять, используя команду **Save** (или нажать клавишу F2) и в открывшемся окне просто щелкнуть по **OK** (или нажать **Enter**).

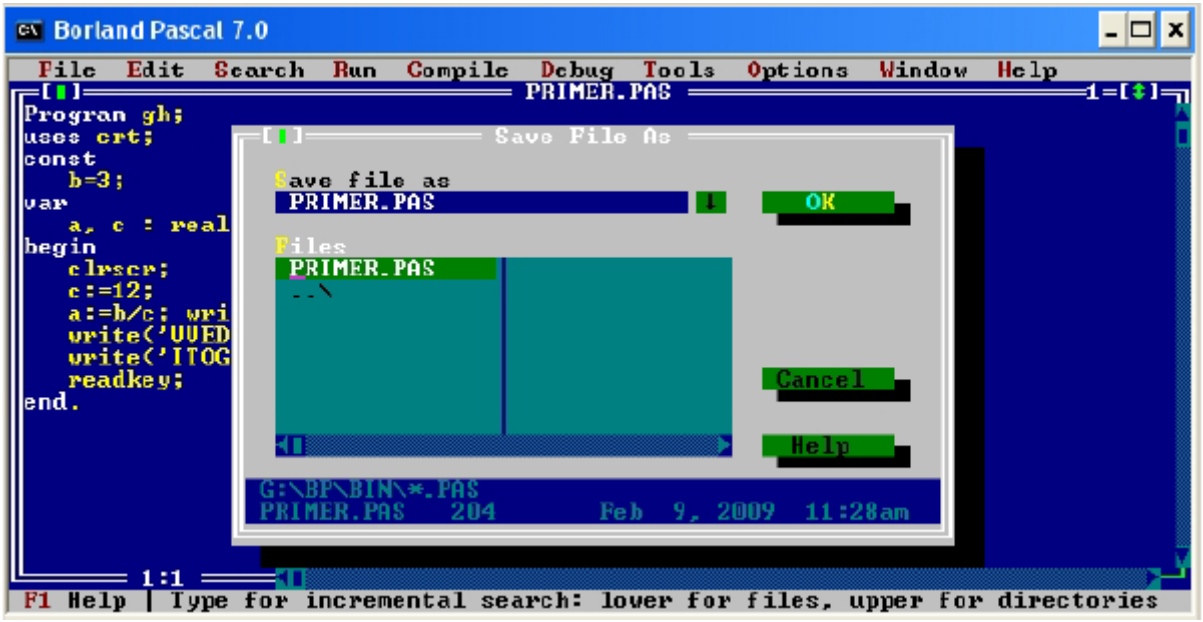

Рисунок 2 – Сохранение программы в среде Турбо Паскаль

6 Контрольные вопросы

1. Что представляют собой современные реализации языка Pascal?
2. Что такое интегрированная интерактивная среда?
3. Перечислите компоненты IDE Turbo Pascal 7.0.
4. Что является средствами IDE Turbo Pascal?
5. Что такое окно? Какие бывают окна?
6. Сколько окон можно открыть в IDE?
7. Перечислите стандартные элементы типичного окна.
8. Как изменить "мышкой" размер окна?
9. Как пользоваться полосами прокрутки?10. Сколько может быть активных окон в IDE Turbo Pascal 7.0, установленной на компьютере с 16 мегабайтами оперативной памяти?
11. Для чего служат окна редактирования?
12. Какие бывают информационные окна?
13. Что произойдет, если нажать на пункт окна меню, после которого стоит стрелочка ?
14. Объясните механизм выбора команд меню с помощью "мышки".
15. Объясните механизм выбора команд меню с помощью клавиатуры.

16. Что произойдет при нажатии комбинации оперативных клавиш Alt+0?

Какие еще оперативные клавиши вы знаете?

17. Перечислите стандартные элементы диалоговых окон?

18. В чем различие кнопок с зависимой и независимой фиксацией?

19. Назовите хотя бы две функции строки состояния.

20. Какая из следующих командных кнопок означает подтверждение: OK, Cancel, Enter, Help, Esc, Ctrl-Alt-Del? Все ли эти кнопки командные?