

2 курс

ПЛАН – КОНСПЕКТ
проведения вводного занятия по теме 2.4
(к практическому занятию № 8)
по дисциплине «Информатика»

Раздел 2. «Общие принципы организации и работы компьютеров.»

Тема № 2.4: «Программное обеспечение персонального компьютера.»

Подготовил: преподаватель
В.Н. Борисов

Рязань 2024

Тема № 2.4. «Программное обеспечение персонального компьютера.»

Цели занятий: изучить со студентами основные сведения о программном обеспечении персонального компьютера.

Основные вопросы:

1. Классификация программного обеспечения (ПО).
2. Базовое программное обеспечение (ПО).
3. Прикладное программное обеспечение (ПО).
4. Назначение, виды, структура систем программирования.
5. Назначение, виды, структура сервисных программ.
6. Назначение, виды, структура программ технического обслуживания.

Литература:

1. [2 учебник раздела «Основной учебной литературы» рабочей программы изучения дисциплины]: Гаврилов, М. В. Информатика и информационные технологии : учебник для среднего профессионального образования / М. В. Гаврилов, В. А. Климов. — 5-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2023. — 355 с. — (Профессиональное образование). — ISBN 978-5-534-15930-1. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/viewer/informatika-i-informacionnyie-tehnologii-510331#page/1>, глава 4.

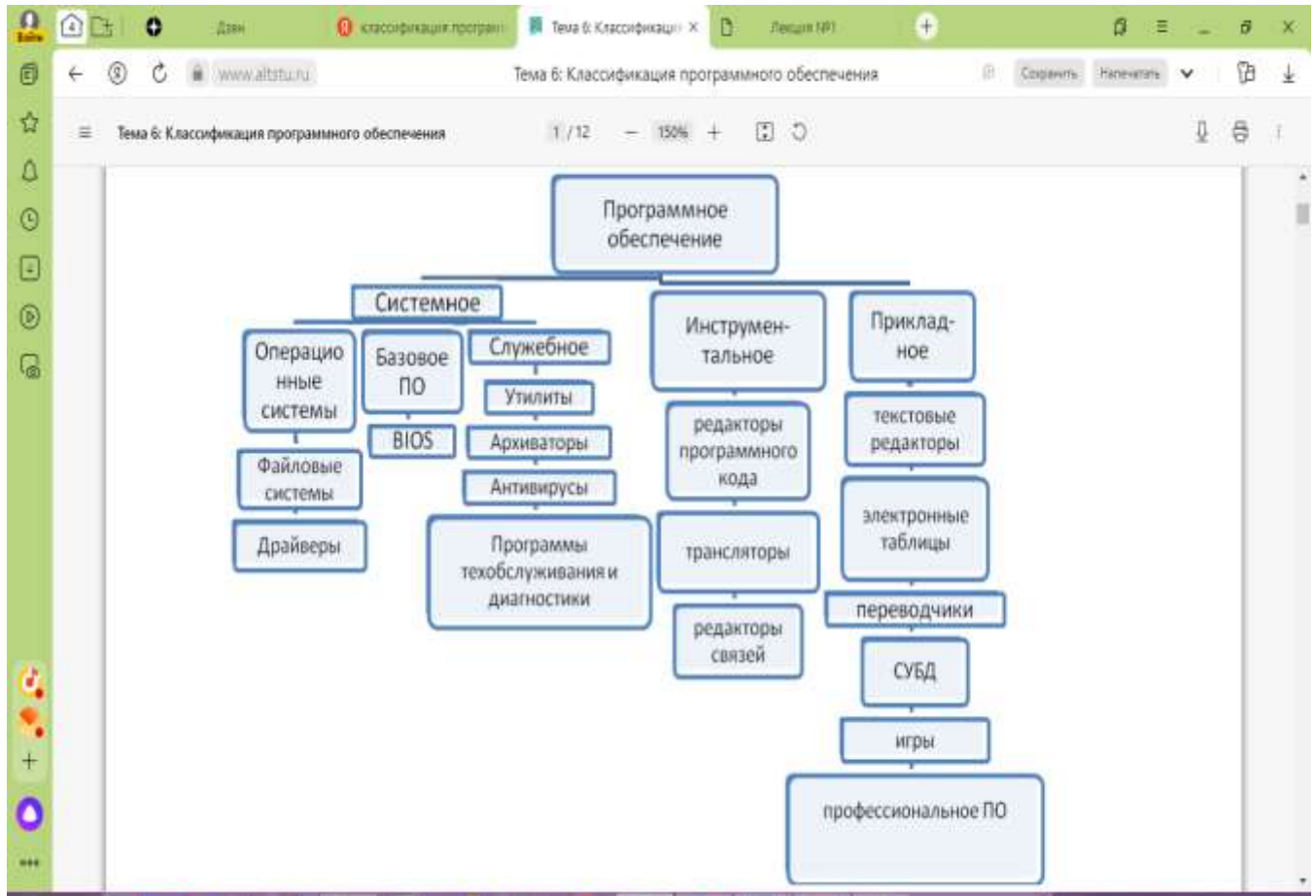
Первый вопрос: Классификация программного обеспечения.

Программное обеспечение (ПО) – неотъемлемая часть компьютерной системы. Оно является логическим продолжением технических средств.

Уровни ПО (снизу вверх):

1. Базовое ПО – базовый уровень
2. Системное ПО – системный уровень.
3. Служебное (сервисное) ПО.
4. Прикладное ПО.

Каждый вышележащий уровень повышает функциональность всей системы.



Традиционно все программное обеспечение подразделяют на два класса:

- 1) системное программное обеспечение (СПО) и
- 2) прикладное (пользовательское) программное обеспечение (ППО)

Выделим еще один класс (скорее группу) программ - **специальное программное обеспечение информационных и управляющих систем.**

Прикладные программы предназначены для решения функциональных задач, они выполняют обработку информации различных предметных областей. Это самый многочисленный класс программных продуктов.

К специальному программному обеспечению информационных и управляющих систем относятся

1. программы (системы) управления базами данных;
2. программы управления языком интерфейса информационных систем;
3. программы сбора и предварительной обработки информации (в информационно-измерительных системах, например, бортовые системы).

ПО этого класса часто оказывается скрытым в составе драйверов оборудования или поставляется в виде библиотек функционального расширения языков программирования.

Поэтому часто такие ПО относят к системному программному обеспечению.

Системное программное обеспечение (System Software) - совокупность программ и программных комплексов для обеспечения работы компьютера и сетей ЭВМ.

СПО управляет ресурсами компьютерной системы и позволяет пользователям программировать в более выразительных языках, чем машинных язык компьютера. Состав СПО мало зависит от характера решаемых задач пользователя.

Назначение системного программного обеспечения.

Системное программное обеспечение предназначено для:

1. создания операционной среды функционирования других программ (другими словами, для организации выполнения программ);
2. автоматизации разработки (создания) новых программ;
3. обеспечения надежной и эффективной работы самого компьютера и вычислительной сети;
4. проведения диагностики и профилактики аппаратуры компьютера и вычислительных сетей;
5. выполнения вспомогательных технологических процессов (копирование, архивирование, восстановление файлов программ и баз данных и т.д.).

Данный класс программных продуктов тесно связан с типом компьютера и является его неотъемлемой частью.

Программные продукты данного класса в основном ориентированы на квалифицированных пользователей - профессионалов в компьютерной области: системного программиста, администратора сети, прикладного программиста, оператора.

Однако знание базовой технологии работы с этим классом программных продуктов требуется и конечным пользователям персонального компьютера, которые самостоятельно не только работают со своими программами, но и выполняют обслуживание компьютера, программ и данных.

Программные продукты данного класса носят общий характер применения, независимо от специфики предметной области.

К системным программным продуктам предъявляются высокие требования по надежности и технологичности работы, удобству и эффективности использования.

Классификация системного программного обеспечения

В СПО традиционно включают:

- 1) системные управляющие
- 2) системные обрабатывающие программы.

Управляющие системные программы организуют корректное функционирование всех устройств системы.

Основные системные функции управляющих программ:

- 1) управление вычислительными процессами и вычислительными комплексами
- 2) работа с внутренними данными ОС.

Как правило, они находятся в основной памяти. Это резидентные программы, составляющие ядро ОС. Управляющие программы, которые загружаются в память непосредственно перед выполнением, называются транзитными (transitive).

В настоящее время системные управляющие программы поставляются фирмами-разработчиками и фирмами-дистрибьюторами в виде инсталляционных пакетов операционных систем и драйверов специальных устройств.

Обрабатывающие системные программы выполняются как специальные прикладные задачи, или приложения.

Эти программы поставляются чаще в виде дистрибутивных пакетов, включающих ПО.

Замечание. В пакеты системных программ помимо основных программ, допускающих реконфигурацию, входят ***специальные настроечные программы***, называемые программами инсталляции.

Другая классификация.

Часто Системное ПО компьютера подразделяют на **БАЗОВОЕ** и **СЕРВИСНОЕ** программное обеспечение.

БАЗОВОЕ программное обеспечение (base software) - минимальный набор программных средств, обеспечивающих работу компьютера.

К базовому программному обеспечению компьютера относятся

1. операционные системы и драйверы в составе ОС;
2. интерфейсные оболочки для взаимодействия пользователя с ОС (операционные оболочки) и программные среды;
3. системы управления файлами.

Системы управления файлами предназначены для организации более удобного доступа к данным, организованным как файлы.

Вместо низкоуровневого доступа к данным с указанием конкретных физических адресов система управления файлами позволяет использовать логический доступ с указанием имени файла.

Любая система управления файлами не существует сама по себе - она разработана для работы в конкретной ОС и с конкретной файловой системой. То есть, можно было бы систему управления файлами отнести к ОС.

Но в связи с тем, что:

- 1) ряд ОС позволяет работать с несколькими файловыми системами (либо с одной из нескольких, либо сразу с несколькими одновременно); а

дополнительную файловую систему можно установить (т.е. они самостоятельны);

2) простейшие ОС могут работать и без файловых систем;

Системы управления файлами выделяются в отдельную группу системных программ.

Часто системы управления файлами относят все-таки к операционным системам.

СЕРВИСНОЕ программное обеспечение – программы и программные комплексы, которые расширяют возможности базового программного обеспечения и организуют более удобную среду работы пользователя.

Это набор сервисных, дополнительно устанавливаемых программ, которые можно классифицировать по функциональному признаку следующим образом:

- 1) драйверы специфических и специальных устройств (те, которые не поставляются в составе ОС);
- 2) программы диагностики работоспособности компьютера;
- 3) антивирусные программы, обеспечивающие защиту компьютера, обнаружение и восстановление зараженных файлов;
- 4) программы обслуживания дисков, обеспечивающие проверку качества поверхности магнитного диска, контроль сохранности файловой системы на логическом и физической уровнях, сжатие дисков, создание страховых копий дисков, резервирование данных на внешних носителях и др.;
- 5) программы архивирования данных, которые обеспечивают процесс сжатия информации в файлах с целью уменьшения объема памяти для ее хранения;
- 6) программы обслуживания сети.

Эти программы часто называются **системными утилитами**. (Заметим, что к антивирусным средствам этот термин обычно не применяется)

Утилиты – программы, служащие для выполнения вспомогательных операций обработки данных или обслуживания компьютеров (диагностики, тестирования аппаратных и программных средств, оптимизации использования дискового пространства, восстановления разрушенной на магнитном диске информации и т.п.).

Отдельную группу системного ПО представляют **системы программирования**.

Второй вопрос: Базовое программное обеспечение.

Базовое ПО – самый низкий уровень ПО. Базовое ПО отвечает за взаимодействие с базовыми аппаратными средствами. Как правило, базовые программные средства непосредственно входят в состав базового оборудования и хранятся в специальных микросхемах, называемых постоянными запоминающими устройствами (ПЗУ). Базовое ПО в архитектуре компьютера занимает особое положение. С одной стороны, его можно рассматривать как составную часть аппаратных средств, с другой стороны, оно является одним из

программных модулей операционной системы. Базовое ПО, или BIOS, представляет программа, которая отвечает за управление всеми компонентами, установленными на материнской плате. Фактически BIOS является неотъемлемой составляющей системной платы и поэтому может быть отнесена к особой категории компьютерных компонентов, занимающих промежуточное положение между аппаратурой и программным обеспечением. Функцией базового программного обеспечения является проверка состава и работоспособности вычислительной системы.

Третий вопрос: Прикладное программное обеспечение.

Это комплекс прикладных программ, с помощью которых на данном рабочем месте выполняются конкретные задания. Это программы конечного пользователя, общего и специализированного назначения. Они предназначены для решения задач в конкретной предметной области.

Классификация прикладных программных средств:

1. Текстовые редакторы
2. Текстовые процессоры
3. Графические редакторы
4. Системы управления базами данных
5. Электронные таблицы
6. Системы автоматизированного проектирования
7. Настольные издательские системы
8. Экспертные системы
9. WEB-редакторы
10. Браузеры
11. Бухгалтерские системы
12. Геоинформационные системы
13. Интегрированные системы делопроизводства
14. Финансовые аналитические системы
15. Системы видеомонтажа

Четвёртый вопрос: Назначение, виды, структура систем программирования.

Система программирования — это система для разработки новых программ на конкретном языке программирования.

Система программирования обычно включает в себя следующие **компоненты**:

1. Компилятор или интерпретатор.
2. Интегрированная среда разработки.
3. Средства создания и редактирования текстов программ.
4. Библиотеки стандартных программ и функций.
5. Отладочные программы, помогающие находить и устранять ошибки.
6. Диалоговая среда.
7. Многооконный режим работы.

8. Мощные графические библиотеки.
9. Утилиты для работы с библиотеками.
10. Ассемблер.
11. Справочная служба.

Компилятор — это особый вид транслятора, который переводит тексты с языка программирования высокого уровня (с того языка, которым пользуется программист при написании текста программы) на машинный язык (в машинный код, который понятен компьютеру).

Интерпретатор — это исполняемый файл, который поэтапно читает программу, а затем обрабатывает, сразу выполняя ее инструкции. Он осуществляет программу поэтапно как часть собственного исполняемого файла.

Интегрированная среда разработки — это набор инструментов для разработки и отладки программ, имеющих общую интерактивную графическую оболочку, поддерживающую выполнение всех основных функций жизненного цикла разработки программы.

Библиотеки стандартных программ и функций состоят из совокупности подпрограмм, составленных на одном из языков программирования и удовлетворяющих определенным единым требованиям к структуре, организации их входов и выходов, описаниям подпрограмм.

Важным компонентом понятия системы программирования являются отладочные программы.

Программный модуль отладки позволяет выполнить основные задачи, связанные с мониторингом процесса выполнения результирующей прикладной программы. Отладка позволяет последовательно и пошагово выполнять итоговые программы, просматривать значения объявленных переменных, устанавливать контрольные точки, трассировку для того, чтобы идентифицировать места и виды ошибок в разработке.

Справочная система, входящая в состав системы программирования, предназначена для предоставления пользователю справочной информации по конкретной системе программирования.

Машинно-ориентированные системы — это системы, в которых язык программирования, наборы операторов и изобразительные средства существенно зависят от особенностей архитектуры компьютера.

Машинно-независимые системы программирования — системы, позволяющие описывать алгоритмы решения задач и информацию, подлежащую обработке. Системы часто используются в широких кругах пользователей и не требуют особых знаний организации функционирования ЭВМ.

Виды языков программирования в машинно-независимых системах:

- процедурно-ориентированные;
- проблемно-ориентированные языки;
- объектно-ориентированное программирование.

Процедурно-ориентированные являются основными языками описания алгоритмов, которые обеспечивают математические функции многих современных вычислительных машин.

Они включают в себя такие популярные языки как:

1. **Fortran** — один из старейших языков программирования высокого уровня, который используется для приложений с интенсивными вычислениями. Fortan часто применяется в процессе научного и инженерного вычисления. Он удобен благодаря большой программной базе, возможностью работы с документами и библиотекам с открытым исходным кодом, доступных под свободными лицензиями. Язык может осуществлять интуитивную запись в виде массива, которая упрощает запись быстрых векторизованных вычислений.
2. **Бейсик** является одним из самых простых языков программирования. Он был создан с целью обучения студентов основам решения задач с помощью написания кода, поэтому программа ориентировалась на пользователей, для которых скорость выполнения программ была не очень важна, и которым первоначально необходима возможность использовать компьютер для решения своих задач, не имея специальной подготовки. В России сегодня наиболее популярна разновидность **Turbo-Basic** фирмы Borland.
3. Язык программирования **Си** был создан как язык высокого уровня для разработки операционной системы **UNIX** и стал популярен благодаря своей простоте и эффективности. Си существенно повлиял на развитие индустрии программного обеспечения; его синтаксис стал основой для современных и востребованных языков **C++**, **C#**, **Java**.
4. **Паскаль** — язык высокого уровня общего назначения, который был первоначально разработан Никлаусом Виртом в начале 1970-х годов. Он разрабатывался с целью создания платформы для **обучения** программированию, поэтому Паскаль недостаточно удобен при решении сложных задач и широко используется для математических операций. Разновидность языка **Паскаль АБС** стала полноценной системой для начинающих, которая сегодня используется для обучения студентов и школьников.

Проблемно-ориентированные языки — это формальные языки, предназначенные для описания данных (информации) и алгоритмов их обработки (программ) на вычислительной машине.

Основные проблемно-ориентированные языки:

1. **ЛИСП** — семейство языков программирования, программы и данные в которых представляются системами линейных списков символов. Так как исходный код состоит из списков, программы на ЛИСПе позволяют его изменять как структуру данных и создавать макросистемы, позволяющие программистам формировать новый синтаксис или новые предметно-ориентированные языки, встроенные в ЛИСП. В настоящее время ЛИСП применяется в экспертных системах, системах аналитических вычислений и т.д.
2. **Prolog** — язык логического программирования, который обеспечивает решение задач, выраженных в терминах объектов и отношений между ними. Для того чтобы инициировать вычисления, выполняется специальный запрос к базе знаний, на которые система логического программирования генерирует ответы «истина» и «ложь».

Объектно-ориентированное программирование основано на методологии представления программы в виде совокупности объектов, каждый из которых является экземпляром определенного класса, а классы образуют иерархию наследования.

Примеры объектно-ориентированных языков:

1. **JavaScript** — язык сценариев, который позволяет создавать интерактивные html-документы, производить вычисления, выполнять проверку допустимости данных без обращения к серверу. Скрипты программы позволяют взаимодействовать с сайтами: заполнять формы обратной связи, оставлять комментарии, просматривать всплывающие подсказки и т.д.
2. **Objective-C** — один из языков программирования, который активно используется для разработки мобильных приложений. Он используется корпорацией Apple и необходим для операционных систем OS X и iOS, их программных интерфейсов.
3. **Python** — высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Он универсален, поэтому подходит для решения разнообразных задач и многих платформ, начиная с iOS и Android и заканчивая серверными ОС. Python используется в различных сферах IT, таких как машинное обучение, разработка приложений, web, парсинг и другие.
4. **Perl** — это язык программирования общего назначения, который изначально был разработан для работы с текстовой информацией, но в дальнейшем стал способен решать широкий круг задач, например, системное администрирование, веб-разработка, сетевое программирование, разработка графического интерфейса пользователя и т.д.

Актуальные системы программирования.

1. **Eclipse** — свободная интегрированная среда разработки модульных кроссплатформенных приложений, которая часто используется как платформа для разработки расширений. Eclipse является платформой с особым фундаментом для построения и запуска интегрированных инструментов разработки сквозного программного обеспечения. В силу бесплатности и высокого качества, Eclipse во многих организациях является корпоративным стандартом для разработки приложений.
2. **IntelliJ IDEA** — мощная универсальная среда программирования, поддерживающая язык **Java**. Она позиционирует себя как умная и удобная система программирования для Java (и других языков) с поддержкой всех последних технологий и фреймворков. У этой среды разработки есть мощные аналитические возможности. Система включает в себя набор инструментов для изменения внутренней структуры программы, который позволяет быстро реорганизовывать исходные тексты. Дизайн среды ориентирован на продуктивность работы программистов и позволяет оптимизировать простые рутинные задачи, чтобы дать возможность специалистам сконцентрироваться на достижение функциональных целей.
3. **Delphi** — среда разработки прикладных программ, предназначенных для запуска в ОС Windows, MacOS, а также в мобильных операционных системах — iOS и Android. Delphi отличается простотой и может использоваться в учебных целях. Программы в Delphi пишутся на языке **Object Pascal**, который является преемником и развитием языка **Turbo Pascal**. Программа предназначена, в первую очередь, для разработки приложений в архитектуре клиент-сервер.
4. Система **Borland C++ Builder** – это мощная среда программирования, которая поддерживает принципы визуального объектно-ориентированного программирования для 32-разрядных операционных систем **Microsoft Windows** и позволяет значительно сократить время на разработку приложений.
5. **Symantec Cafe** — первая интегрированная среда визуальной разработки для создания приложений и интернет-страниц. Symantec Cafe позволяет разрабатывать приложения на языке Java, которые могут затем встраиваться в интернет-страницы для повышения их функциональности.
- 6.

Пятый вопрос: Назначение, виды, структура сервисных программ.

Пользовательский интерфейс (сервисные программы) – это программные надстройки операционной системы (оболочки и среды), предназначенные для упрощения общения пользователя с операционной системой.

Сервисные программы предназначены для выполнения различных вспомогательных операций – проверки исправности оборудования, архивации файлов, борьбы с вирусами, форматирования дисков (подготовки новых дисков к работе путем разметки на них дорожек и секторов).

Программы, обеспечивающие интерфейс, сохраняют форму общения (диалог) пользователя с операционной системой, но изменяют язык общения (обычно язык команд преобразуется в язык меню). Сервисные системы условно можно разделить на интерфейсные системы, оболочки операционных систем и утилиты.

Интерфейсные системы – это мощные сервисные системы, чаще всего графического типа, совершенствующие не только пользовательский, но и программный интерфейс операционных систем, в частности, реализующие некоторые дополнительные процедуры разделения дополнительных ресурсов.

Оболочки операционных систем предоставляют пользователю качественно новый по сравнению с реализуемой операционной системой интерфейс и делают необязательным знание последнего.

Утилиты автоматизируют выполнение отдельных типовых, часто используемых процедур, реализация которых потребовала бы от пользователя разработки специальных программ. Многие утилиты имеют развитый диалоговый интерфейс с пользователем и приближаются по уровню общения к оболочкам.

Утилиты предназначены для расширения возможностей операционной системы и встроенных в систему служебных программ за счет введения новых или усовершенствования уже существующих функций.

Так компьютерные утилиты можно разделить на три группы:

- 1) утилиты сервисного обслуживания компьютера,
- 2) утилиты расширения функциональности,
- 3) информационные утилиты.

Многие утилиты дублируют друг друга. Их перечень очень широк, но по функциональному назначению их разделяют на несколько групп:

- 1) антивирусные программы;
- 2) файловые менеджеры;
- 3) программы резервирования;
- 4) программы-упаковщики (архиваторы);
- 5) программы контроля и диагностики компьютера;
- 6) программы обслуживания дисков;
- 7) программы – КЭШи и программы оптимизации дисков;
- 8) программы обслуживания операционной системы....

Шестой вопрос: Назначение, виды, структура программ технического обслуживания.

Под программами технического обслуживания понимается совокупность программно-аппаратных средств для диагностики и обнаружения ошибок в процессе работы компьютера или вычислительной системы в целом.

Они включают в себя:

- средства диагностики и тестового контроля правильности работы ЭВМ и ее отдельных частей, в том числе автоматического поиска ошибок и неисправностей с определенной локализацией их в ЭВМ;
- специальные программы диагностики и контроля вычислительной среды информационной системы в целом, в том числе программно-аппаратный контроль, осуществляющий автоматическую проверку работоспособности системы обработки данных перед началом работы вычислительной системы в очередную производственную смену.

Комплекс программ технического обслуживания

Особенности состава технических средств ЭВМ учитываются комплексом программ технического обслуживания (КПТО). Этот комплекс включает в свой состав наладочные, проверочные и диагностические тест-программы.

Наладочные программы обеспечивают автономную настройку и проверку отдельных устройств ЭВМ. Обычно они функционально независимы от программ ОС. Проверочные тест-программы предназначены для периодически проводимых проверок правильности функционирования устройств, например после включения их в работу. Диагностические программы используются в тех случаях, когда необходимо классифицировать отказ оборудования и локализовать место неисправности. Инициирование работы этих программ осуществляется обычно модулями ОС после фиксации сбоев и отказов аппаратуры контроля.

Проверочные тест – программы занимают особое место в КПТО. Их выполнение непосредственно перед вычислениями позволяет убедиться в исправности технических средств системы, а значит, повысить достоверность результатов обработки данных.

Кроме встроенных средств контроля, в ПО ПЭВМ включаются и автономные средства контроля и диагностики. Количество подобных комплектов программ достаточно велико, и каждый из них позволяет детализировать системную информацию: определение полной конфигурации ПК и характеристик отдельных ее частей (тип процессора, наличие сопроцессора, тип материнской платы, типы используемых дисков, объем оперативной памяти и ее распределение, подключение дополнительной периферии и т.д.).

Заключительная часть (по каждому занятию).

1. Закончить изложение материала.
2. Ответить на возникшие вопросы.
3. Подвести итоги занятий.
4. Дать задания на самоподготовку (домашние задание).

Задание на самоподготовку (домашние задания):

1. Детально проработать, законспектировать материал занятий, в том числе размещенный в учебнике, указанном на с.2 данного план-конспекта.
2. Подготовиться к опросу по пройденному материалу, защите ранее выполненных практических работ.