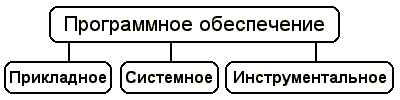
|  |
| --- |
| **Содержание**  **Введение** **1**  Основные определения ПО 2  Виды классификации ПО по назанчению 3  Система програмирования 4  Этапы подготовки программы при работе 4  Разница между языками 4  **Заключение** **5**  **Список литературы 6**  Приложение 7 |

**Основные определения ПО**



|  |
| --- |
| Сам по себе компьютер не обладает знаниями ни в одной области применения. Все эти знания сосредоточены в выполняемых на компьютерах программах. |

**Програ́ммное обеспе́чение** (**ПО**) — программа или множество программ, используемых для управления компьютером.

Имеются и другие определения из международных и российских стандартов:

* совокупность программ системы обработки информации и программных документов, необходимых для эксплуатации этих программ
* все или часть программ, процедур, правил и соответствующей документации системы обработки информации;
* компьютерные программы, процедуры и, возможно, соответствующая документация и данные, относящиеся к функционированию компьютерной системы.

Программное обеспечение является одним из видов обеспечения вычислительной системы, наряду с техническим (аппаратным), математическим, информационным, лингвистическим, организационным, методическим и правовым обеспечением.

Академические области, изучающие программное обеспечение, — это информатика и программная инженерия.

**Виды классификации ПО по назанчению**

**В зависимости от назначения, а также области действия, принято выделять три основных класса ПО:**

**Системное (системные программы)** - предназначены для эксплуатации и технического обслуживания ЭВМ, управления и организации вычислительного процесса при решении любой конкретной задачи на ЭВМ и т.д.

К ним относятся:

* операционные системы;
* оболочки операционных систем;
* драйверы;
* программы-утилиты и программы технического обслуживания;
* антивирусные программы и др;

**Прикладное (системы программирования)**

**Инструментарий для программирования (инструментальные программы)** - это программы, используемые как инструмент при создании документов в повседневной деятельности. Как правило это не отдельные программы, а пакеты программ. К ним относятся:

* программы для работы с текстами (текстовые редакторы);
* графические редакторы;
* электронные таблицы (табличный процессор);
* системы управления базами данных (СУБД);
* программы подготовки презентаций;
* программы для Web-дизайна;
* программы распознавания символов;
* интегрированные среды.



**Система програмирования**

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

Современные системы программирования обычно предоставляют пользователям **мощные и удобные средства разработки программ.** В них входят:

* компилятор или интерпретатор;
* интегрированная среда разработки;
* средства создания и редактирования текстов программ;
* обширные библиотеки стандартных программ и функций;
* отладочные программы, т.е. программы, помогающие находить и устранять ошибки в программе;
* "дружественная" к пользователю диалоговая среда;
* многооконный режим работы;
* мощные графические библиотеки; утилиты для работы с библиотеками
* встроенный ассемблер;
* встроенная справочная служба;
* другие специфические особенности.

**Этапы подготовки программы при разработке**

* Этап 1 – Определение проблемы
* Этап 2 – Выработка требований
* Этап 3 – Создание плана разработки
* Этап 4 – Разработка архитектуры системы или высокоуровневое проектирование
* Этап 5 – Детальное проектирование
* Этап 6 – Кодирование и отладка
* Этап 7 – Тестирование компонентов
* Этап 8 – Интеграция компонентов
* Этап 9 – Тестирование всей системы
* Этап 10 – Сопровождение, внесение изменений, оптимизация

**Этап 1 – Определение проблемы**

Перед тем как приступать к кодированию, необходимо четко сформулировать проблему, которую Ваша будущая программа должна решать. Так как, не имея хорошего определения проблемы, Вы можете потратить много усилий и времени на решение не той проблемы, которую требуется решить.

На данном этапе проводится простая формулировка сути проблемы без каких-либо намеков на ее возможные решения, при этом формулировать ее следует на языке, понятном пользователю, т.е. она должна быть описана с пользовательской точки зрения.

**Этап 2 – Выработка требований**

Что такое требования и зачем их нужно выработать?

**Требования к программе** – это подробное описание всех возможностей программы и действий, которые должна выполнять программа. Такие требования иногда также называют *«Функциональной спецификацией»* или просто *«Спецификацией»*.

Требования вырабатывают для того, чтобы свести к минимуму изменения системы после начала непосредственной разработки. Такие требования должны быть обязательно официальными, т.е. документально оформлены. Так как это гарантирует то, что функциональность системы определяется заказчиком, а не программистом. Даже в случае с внутрикорпоративными разработками такие требования должны быть зафиксированы, например, в виде технического задания, подписанного всеми задействованными лицами, тем самым Вы избежите лишних разговоров и споров, например, о том, что реализованный функционал делает не все или не так.

Выработка требований очень важна, так как она позволяет определить функциональность программы до начала программирования.

**Этап 3 – Создание плана разработки**

На данном этапе Вы уже должны в формальном виде составить план разработки программного обеспечения с учётом существующей проблемы и выработанных требований. Иными словами, Вы должны составить план того, как Вы будете действовать дальше.

**Этап 4 – Разработка архитектуры системы или высокоуровневое проектирование**

Данный этап также очень важный, так как, не имея хорошей архитектуры, Вы можете решать правильную проблему, но прийти к неправильному решению. Хорошая архитектура программы упрощает программирование, а плохая архитектура усложняет его.

Архитектура системы обычно включает:

* Общее описание системы;
* Основные компоненты;
* Формат и способ хранения данных;
* Специфические бизнес-правила;
* Способ организации пользовательского интерфейса;
* Подход к безопасности системы;
* Оценки производительности;
* Возможности масштабирования;
* Моменты, связанные с интернациональностью, т.е. будет ли система интернациональной.

Кроме того, в архитектуру необходимо включить подтверждение того, что при разработке этой архитектуры рассматривались альтернативные варианты в каждом из вышеперечисленных направлений, с обоснованием окончательного выбора и подхода.

**Этап 5 – Детальное проектирование**

На этом этапе проводится проектирование программы на низком уровне, иными словами, здесь проектируются классы и методы, рассматриваются, оцениваются и сравниваются различные варианты и причины выбора окончательных подходов и способов реализации.

При разработке небольших программ программисты обычно сами проектируют программу на таком уровне, это выглядит как написание псевдокода или рисование схем, поэтому часто этот этап рассматривается как часть непосредственного кодирования и в таких случаях итоговый документ (если того требует формальность) состоит преимущественно из различных набросков и заметок программистов.

Но при реализации крупных проектов данному процессу отводится отдельный этап и проектирование в этом случае проводится с очень высокой степенью детальности.

**Этап 6 – Кодирование и отладка**

Это как раз тот этап, который все знают и, наверное, думают, что это единственный этап в процессе разработке программного обеспечения – это непосредственное написание кода и его отладка. Но, как видите, это далеко не первый и не единственный этап разработки ПО.

Если все вышеперечисленные этапы выполнены, то данный этап подразумевает чисто механическую работу, т.е. кодинг. Программисту в этом случае не нужно что-то выдумывать и самостоятельно разрабатывать, ему нужно просто написать код, который реализует заданный, очень детально описанный в проекте, алгоритм.

После того как код написан, программисту необходимо отладить этот код, чтобы в нем не было никаких ошибок.

**Этап 7 – Тестирование компонентов**

После того, как код написан, и проведена отладка, необходимо провести тестирование реализованного функционала. Если программа состоит из нескольких компонентов, сначала тестируют каждый компонент в отдельности, так как очень крупные программы включают огромный функционал, который часто разделяют на отдельные компоненты, разработка которых осуществляется по отдельности. В менее крупных проектах этот этап может включать просто тестирование отдельных классов.

**Этап 8 – Интеграция компонентов**

Когда тестирование всех компонентов закончено, можно переходить к интеграции всех компонентов в единый программный комплекс, этот этап как раз и подразумевает процесс интеграции, т.е. слияния всех компонентов в единую систему.

В небольших проектах этот этап может заключаться в объединении нескольких классов, на что будет затрачено не больше одного дня, но в крупных проектах этот этап может длиться не один месяц.

**Этап 9 – Тестирование всей системы**

На данном этапе проводится тестирование всей системы, уже с учётом интеграции всех компонентов. На этом этапе можно выявить проблемы взаимодействия компонентов и устранить их. Также на этом этапе основным предметом тестирования является безопасность, производительность, утечка ресурсов и другие моменты, которые невозможно протестировать на более низких уровнях тестирования.

**Этап 10 – Сопровождение, внесение изменений, оптимизация**

После запуска программы в промышленную эксплуатацию осуществляется сопровождение этой программы, т.е. внесение изменений на основе выявленных недочетов в процессе эксплуатации системы, а также проводится оптимизация функционала или добавление нового.

**Ссылки:**

1. <http://book.kbsu.ru/theory/chapter6/1_6.html>
2. [Ссылка](https://unix-it.ru/information/klassifikatsiya-programmnogo-obespecheniya-kompyutera/#:~:text=%D0%92%20%D0%B7%D0%B0%D0%B2%D0%B8%D1%81%D0%B8%D0%BC%D0%BE%D1%81%D1%82%D0%B8%20%D0%BE%D1%82%20%D0%BD%D0%B0%D0%B7%D0%BD%D0%B0%D1%87%D0%B5%D0%BD%D0%B8%D1%8F%2C%20%D0%B0,%D0%B8%D0%BD%D1%81%D1%82%D1%80%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%80%D0%B8%D0%B9%20%D0%B4%D0%BB%D1%8F%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F.)
3. [Система програмирования](http://book.kbsu.ru/theory/chapter6/1_6_11.html)
4. [Этапы подготовки программы при разработке](https://info-comp.ru/programmirovanie/724-stages-of-program-development.html)