

# Основные понятия и определения

**Программа** – упорядоченная в соответствии с некоторым алгоритмом последовательность команд (инструкций) компьютера для решения задачи пользователя.

**Специфика разработки программных систем (ПС):**

- Неформальный характер требований к ПС (постановки задачи) противопоставлен формальному понятию ошибки в ПО.
- Разработка ПС носит творческий.
- ПС при своем использовании (эксплуатации) не расходуется и не расходует используемых ресурсов (позиция, на которую опираются пираты).

**Технология программирования** (programming technology) – совокупность производственных процессов, приводящая к созданию требуемой ПС, а также описание этой совокупности процессов (акцент на процессах разработки ПС).

**Программная инженерия** – систематический подход к разработке, эксплуатации, сопровождению и изъятию из обращения программных (акцент на методах и инструментальных средствах).

# Основные понятия и определения

- **Технология программирования:** методы рассматриваются "сверху" – с точки зрения организации технологических процессов.
- **Методология программирования:** методы рассматриваются "снизу" – с точки зрения основ их построения.

# Основные понятия и определения

**Программная инженерия** = правила инженерной разработки ПС.

Ядро знаний **SWEBOK** (Software Engineering Body of Knowledge) (2001, 2003 гг.) – систематизированы знания в области программирования, планирования и управления, сформулировано понятие программной инженерии и областей, которые соответствуют процессам проектирования ПС и методам их поддержки.

Ядро знаний **SWEBOK** создано Международным комитетом при американском объединении компьютерных специалистов ACM (Association for Computing Machinery) и институте инженеров по электронике и электротехнике IEEE

# Этапы развития технологий программирования

## «Стихийное» программирование

- Период от появления первых ЭВМ до середины 60-х гг. 20-го века;
- Отсутствие сформулированной технологии (программирование = искусство);
- Машинные языки → ассемблеры → алгоритмические языки (Fortran, Algol)
- Концепция повторного использования подпрограмм, что повысило производительность труда программиста.
- Разработка "снизу вверх" – вначале проектировали и реализовали сравнительно простые подпрограммы, из которых потом строилась сложную программу.
- Итог – кризис программирования – фирмы срывали дедлайны и бюджеты.

# Этапы развития технологий программирования

## Структурный подход к программированию

- Период начался в 60-70-е годы 20-го века и представлял собой;
- Совокупность рекомендуемых технологических приемов на всех этапах разработки;
- Декомпозиция – представление задачи в виде иерархии подзадач (в отличие от процедурного подхода, где строился набор подпрограмм).
- Разработка "сверху вниз" = реализация общей идеи, обеспечивая проработку интерфейсов подпрограмм;
- Метод проектирования алгоритмов – метод пошаговой детализации.
- Процедурные языки программирования (PL/1, Algol-68, Pascal, C)

# Этапы развития технологий программирования

## **Структурный подход к программированию**

**Технология модульного программирования** – выделение групп подпрограмм, использующих одни и те же глобальные данные, в отдельно компилируемые модули.

Структурный подход + модульное программирование = **надежные программы**, размер которых не превышает 100000 операторов.

**Узким местом** модульного программирования стали межмодульные интерфейсы, ошибки в которых трудно обнаружить по причине отдельной компиляции модулей (ошибки выявляются только при выполнении программы).

# Этапы развития технологий программирования

## Объектный подход к программированию

- Период с середины 80-х до конца 90-х годов 20-го века;
- **Объектно-ориентированное программирование (ООП)** = технология создания сложного ПО, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определенного типа (класса), а классы образуют иерархию с наследованием свойств. Взаимодействие программных объектов осуществляется путем передачи сообщений.
- **Основное достоинство ООП** по сравнению с модульным программированием – более естественная декомпозиция программного обеспечения, которая существенно облегчает его разработку.
- Способы организации программ, основанные на механизмах **наследования, полиморфизма, композиции**, позволяет существенно увеличить показатель повторного использования кодов и создавать библиотеки классов для различных применений.
- Создание сред **визуального программирования**.
- Языки визуального объектно-ориентированного программирования (Delphi, C++ Builder, Visual C++, C# и т. д.)

# Этапы развития технологий программирования

## Компонентный подход и CASE-технологии

- С середины 90-х годов 20-го века до нашего времени
- Построение ПО из **отдельных компонентов** – физически отдельно существующих частей программного обеспечения, которые взаимодействуют между собой через стандартизированные интерфейсы.
- В отличие от обычных объектов **объекты-компоненты** можно собирать в динамически вызываемые библиотеки или исполняемые файлы, распространять в двоичном виде (без исходных текстов) и использовать в любом языке программирования, поддерживающем соответствующую технологию.
- Основы компонентного подхода были разработаны компанией Microsoft, начиная с **технологии OLE** (Object Linking and Embedding – связывание и внедрение объектов), которая применялась в ранних версиях Windows для создания составных документов. Ее развитием стало появление **COM-технологии** (Component Object Model – компонентная модель объектов), а затем ее распределенной версии – DCOM, на основе которых были разработаны компонентные технологии, решаются различные задачи разработки программного обеспечения



# Этапы развития технологий программирования

## **Компонентный подход и CASE-технологии**

**CASE-технологии** (Computer Aided Software/System engineering – разработка программного обеспечения/программных систем с использованием компьютерной поддержки) – важнейшая особенность современного этапа технологии программирования – широкое использование компьютерных технологий создания и сопровождения программных систем на всех этапах их жизненного цикла.

# Качество и характеристики программного обеспечения

- **Качество ПО** – это совокупность свойств, определяющих полезность изделия (программы) для пользователей в соответствии с функциональным назначением и предъявленными требованиями.
- **Характеристика качества программы** отражает отдельные факторы, влияющие на качество программ и поддающиеся измерению .
- **Критерий качества ПО** – численный показатель, характеризующий степень, в которой программе присуще оцениваемое свойство.

# Качество и характеристики программного обеспечения

**Критерии качества** могут включать множество различных характеристик:

- экономичность,
- документированность,
- гибкость,
- модульность,
- надежность,
- обоснованность,
- тестируемость,
- ясность,
- точность,
- модифицируемость,
- эффективность,
- легкость сопровождения
- и т.д.

•**Метрика качества программ** – это система измерений качества программ.

# Качество и характеристики программного обеспечения

**Критерии качества ПС** (criteria of software quality):

- **функциональность (обязательный)** – способность ПС выполнять набор функций, удовлетворяющих заданным потребностям пользователей;
- **надежность (обязательный)** – устойчивость, точность выполнения предписанных функций обработки, возможность диагностики возникающих ошибок.

**Надежность (reliability) ПС** – это ее способность безотказно выполнять определенные функции при заданных условиях в течение заданного периода времени с достаточно большой вероятностью.

**Отказ в ПС** – проявление в ней ошибки.

**Надежная ПС** не исключает наличия в ней ошибок – важно лишь, чтобы эти ошибки при практическом применении этого ПС в заданных условиях проявлялись достаточно редко. Убедиться, что ПС обладает таким свойством, можно при ее испытании путем тестирования, а также при практическом применении;

# Качество и характеристики программного обеспечения

**Критерии качества ПС** (criteria of software quality):

- **легкость применения** – минимальные усилия по подготовке исходных данных, дружелюбный интерфейс, хорошая документация;
- **эффективность** – это отношение уровня услуг, предоставляемых ПС пользователю при заданных условиях, к объему используемых ресурсов;
- **сопровождаемость, модифицируемость** – минимизировать усилия по внесению изменений для устранения в ней ошибок и по ее модификации в соответствии с изменяющимися потребностями пользователей, переходу на новые версии и т.п.;
- **мобильность** (кроссплатформенность) – независимость от технического комплекса вычислительных средств, операционной системы, сетевых возможностей, специфики предметной области задачи и т. д.;
- **коммуникативность** – степень возможной интеграции с другими программами, обеспечение обмена данными между программами.

# Качество и характеристики программного обеспечения

- Процесс разработки ПС и методы оценивания продуктов **стандартизованы** (ISO/IEC 12207, 9126 и др.) - это способствует повышению эффективности проектирования, разработки, тестирования и оценки качества ПС.

# Классы программного обеспечения

Выделяют три класса программных продуктов:

- **системное программное обеспечение** (System Software) – совокупность программ и программных комплексов, предназначенная для обеспечения работы компьютера и сетей ЭВМ;
- **прикладное программное обеспечение** – комплекс взаимосвязанных программ, предназначенный для решения задач определенного класса конкретной предметной области;
- **инструментальное программное обеспечение** – совокупность программ и программных комплексов, обеспечивающих технологию разработки, отладки и внедрения создаваемых программных продуктов.

# Классы программного обеспечения

