

# Классификация ПП.

Критерий - сфера (область) использования ПП.

## *1. Системное программное обеспечение*

Предназначено для выполнения различных вспомогательных функций:

- управление ресурсами вычислительной машины;
- проверку работоспособности устройств и др.

## Примеры:

- **Операционные системы**

(управляют всеми элементами компьютера и взаимодействием программного обеспечения и аппаратуры);

- **Антивирусные программы;**

- **Программы архивирования;**

- **Программы обслуживания абонентов сети и др.**

## ***2. Прикладные программы (системы).***

(Ориентированы на пользователя не программиста в некоторой предметной области).

### **Примеры:**

- ✓ Редакторы (Microsoft Word);
- ✓ Системы машинной графики (научные, инженерные и др.);
- ✓ Электронные таблицы (Microsoft Excel);
- ✓ Системы управления базами данных (Microsoft Access);
- ✓ Издательские системы; Aldus PageMaker, Corel Ventura.
- ✓ Бухгалтерские программы (1С Бухгалтерия);
- ✓ Системы автоматизированного проектирования;
- ✓ Системы искусственного интеллекта;
- ✓ Игровые системы.
- ✓ Оболочки прикладных систем и др.

### *3. Инструментальные средства создания ПП*

(Облегчают процесс создания новых программ (систем) на конкретном языке программирования с использованием библиотек стандартных компонентов.)

### *4. Гибридные системы*

(Сочетают в себе элементы вышеприведенных систем)

Классификация с точки зрения вида обрабатываемой информации, **например:**

- Системы обработки текстовой информации;
- Системы обработки графической информации;
- Системы обработки аудиоинформации.

Также системы могут делиться на **сетевые** и **не сетевые** (сосредоточенные и распределенные) и др.

## Системы редактирования

Можно выделить ряд общих функций:

- Функции для работы с файлами (создать, открыть, сохранить, распечатать и др.);
- Просмотр и редактирование в режиме диалога (основной режим, используются непосредственно группы клавиш - редактирования, алфавитно-цифровые и др.);
- Функции для работы с фрагментами (переместить, копировать, удалить, распечатать и др.);

- **Для редакторов текстов** отдельно можно выделить, например, функцию контекстного поиска подстроки (с заменой).
- **В редакторах текстов программ** должны быть: функция проверки на наличие синтаксических ошибок, подцветка основных элементов и др.
- **В редакторах документов** — использование различных шрифтов, вставка страниц, выравнивание текста и др.
- **В графических редакторах** — функции для работы с фотодокументами и для выполнения чертежных, цветных, художественных работ.
- **В издательских системах** необходима автоматизация процесса верстки, при этом нужно высокое качество и вывод на фотонаборные автоматы.
- **В редакторах научных документов** возможность быстро набирать математические, химические и др. формулы.

# Системы баз данных

## Основные функции:

- Хранение данных;
- Обеспечение доступа к БД;
- Защита целостности БД при аппаратных и программных сбоях;
- Защита от несанкционированного доступа;
- Обновление данных;
- Сортировка и индексирование данных;
- Поиск и отбор данных;
- Формирование отчетов;
- Управление транзакциями, т.е. последовательностями операций над БД;
- Журнализация и восстановление БД после сбоев и др.

# Интеллектуальные системы.

Моделируют разумную деятельность человека.

Основные черты:

- ✓ - возможность накапливать и получать новые знания;
- ✓ способность осуществлять рассуждения и получать объяснения по ним;
- ✓ осуществлять семантический диалог с человеком.

**К таким системам можно отнести:**

обучающие системы, системы контроля знаний, экспертные системы, лингвистические системы и др.

Основное назначение экспертных систем вытекает из определения:

*«ЭС - компьютерная программа, которая ведет себя подобно человеку-эксперту в некоторой проблемной области».*



## Гипертекстовые системы.

(Позволяют автоматизировать процесс получения нового знания, усиливает творческие возможности человека.)

### Три основные отличительные функции:

- установление различных связей между элементами информационного массива;
- поиск информации в созданной ранее ГТ БД;
- добавление новых узлов и связей в ГТ БД.

**ГТ база данных** - совокупность текстов и других источников информации, между выделенными элементами которых установлены различные связи (неявные, явные)

**ГТ документ** - это документ, который получен в результате прохождения по маршруту вдоль установленных ранее связей. При выходе из системы такой документ становится обычным линейным текстом.

# История развития средств и технологий разработки ПП.

## 1. «Стихийное» программирование

(от первых ВМ 1949г. до середины 60-х годов)

- ✓ Технология практически отсутствует.
- ✓ Первые программы и данные составлялись на машинном языке.
- ✓ После появления ассемблеров вместо двоичных кодов используются символические имена данных и мнемоники кодов операций.

### Проблемы:

- ✓ отсутствует наглядность программы;
- ✓ большая вероятность сделать ошибку;
- ✓ нужно хорошо знать и представлять структуру компьютера, взаимосвязь его блоков и элементов.

Появились языки высокого уровня FORTRAN и ALGOL.

- ✓ Упростилось программирование вычислений.
- ✓ Увеличилась сложность программ.

## Программа состояла из:

основной программы;  
области глобальных данных;  
набора подпрограмм.

Появилась возможность создавать библиотеки.



**Недостаток** - при увеличении количества подпрограмм возрастала вероятность искажения части глобальных данных какой-либо подпрограммой.

**Предложено** в подпрограммах размещать **локальные данные**.

**Стало возможным** распараллеливать процесс разработки между несколькими программистами.

**В начале 60-х случился «кризис».**

- ✓ При разработке сложного ПП срывались сроки выполнения;
- ✓ Проект устаревал раньше, чем был готов к внедрению;
- ✓ Увеличивалась стоимость проекта;
- ✓ Тестирование и отладка занимали более **???** % времени разработки.

## 2. Структурное программирование (60-е –70-е годы)

В основе лежит процедурная декомпозиция  
(сложная система разбивается на отдельные небольшие подпрограммы (представляются в виде иерархии)).

Данный подход поддерживали *процедурные языки*, например:  
PL/1, ALGOL-68, Pascal, C и др.

Далее развивается структурирование данных (появляется возможность определения пользовательских типов данных.)

В результате появилась технология *модульного программирования*, которую поддерживали Pascal и C (C++), языки Ада и Modula.

*Недостатки:* - из-за отдельной компиляции модулей ошибка в интерфейсе выявляется **только** при выполнении всей программы и дополнительные затраты на межмодульный интерфейс.

### 3. Объектно-ориентированное программирование ( с середины 80-х до нашего времени)

**Программа в виде совокупности *объектов*.**

#### Основные достоинства:

- сократилось количество межмодульных вызовов;
- уменьшился объем информации, передаваемой между модулями.

Далее появились среды **визуального** программирования.

(Результатом визуального проектирования является **полуфабрикат будущей программы**, созданный на основе стандартных элементов.)

# Появились средства автоматизирующие проектирование и разработку ПО:

## CASE (Computer-Aided Software/System Engineering)

(с приходом стандарта ISO/IEC 14102 CASE-средства стали использовать как программные средства для поддержки процессов жизненного цикла ПО.)

### Примеры CASE-инструментов:

- инструменты **управления конфигурацией**;
- инструменты **моделирования данных**;
- инструменты **анализа и проектирования**;
- инструменты **преобразования моделей**;
- инструменты **редактирования программного кода**;
- инструменты **рефакторинга кода** (процесс изменения внутренней структуры программы с целью повышения технологичности за счет стиля программирования).
- **генераторы кода**;
- инструменты для **построения UML-диаграмм**.

# UML (Unified Modeling Language) - унифицированный язык моделирования.

Средство моделирования для построения:

- концептуальных
- логических
- графических

моделей сложных систем широкого назначения.

**В частности,** основывается на понимании общих принципов моделирования объектно-ориентированного проектирования (абстрагирование и др.).

Цель – отобразить все особенности моделируемой системы, касающиеся как статического, так и динамического представления.

\*\*\* При этом используется принцип иерархического построения моделей сложных систем.



Создание UML началось в конце 1994 г.

UML является объединением и унификацией методов Гради Буча, Джеймса Рамбо и Ивара Якобсона и плюс дополнительные возможности.

### В итоге пользователям предоставляется:

- ✓ язык визуального моделирования;
- ✓ механизмы расширяемости и специализации;
- ✓ обеспечивается независимость от конкретных языков программирования и процессов разработки и др.

**<https://www.uml-diagrams.org/uml-25-diagrams.html>**