

# Разработка программного модуля. Технологии программирования.

---

**Отвагин Алексей Владимирович,  
доцент каф. ЭВМ, к.т.н., а. 505-5**

# Содержание

---

- Порядок разработки программного модуля
- Структурное программирование
- Понятие псевдокода
- ООП
- Компонентный подход
- Параллельное программирование
- Агентное программирование

# Общий порядок разработки программного модуля

---

- Изучение и проверка спецификации
- Выбор языка программирования, алгоритма и структур данных
- Кодирование
- Проверка модуля
- Компиляция

# Изучение и проверка спецификации

---

- Выполняется смежным контролем – изучением спецификации ее разработчиком и кодировщиком модуля
- Этап предполагает уточнение спецификации и решения задачи

# Выбор ЯП, алгоритма и структур данных

---

- Начинается с выбора алгоритма решения задачи
- Выделяются основные сущности и отношения между ними – структуры данных
- Анализируются готовые решения в виде библиотек или конструкций языка

# Кодирование

---

- Построение текста программы на выбранном языке
- Требует использования определенной дисциплины программирования – набора правил и устойчивых конструкций
- Управляется стандартами кодирования

# Проверка и компиляция

---

- Внутренняя проверка логики модуля и его соответствия спецификации
- Документирование модуля в соответствии со стандартами оформления (комментарии, описание входов-выходов и др.)
- Компиляция является завершающим этапом для перехода к отладке

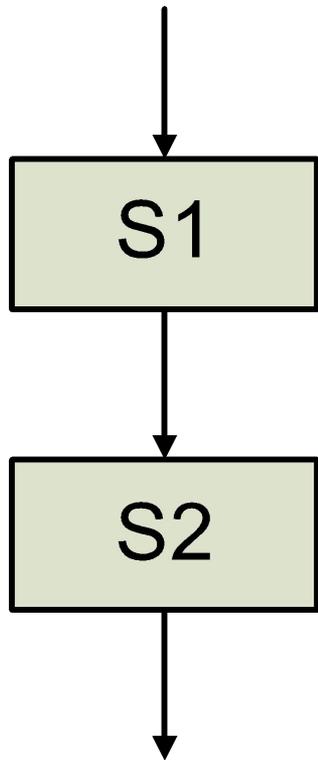
# Структурное программирование

---

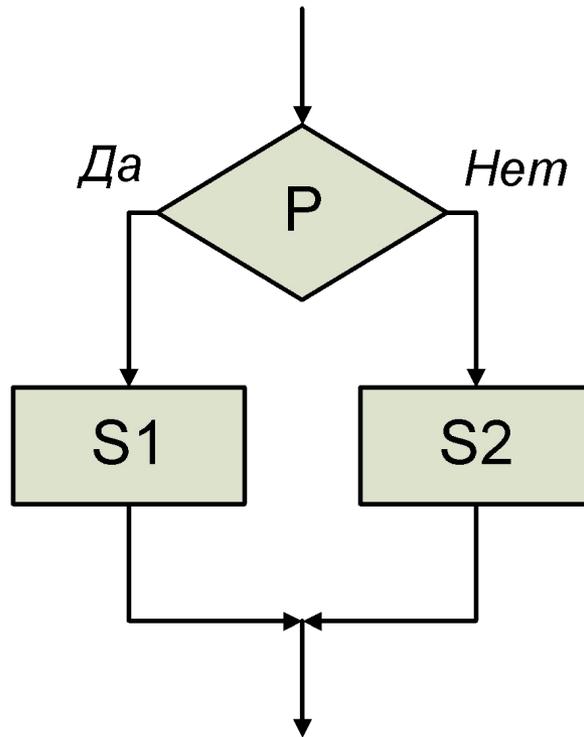
- Первая систематизированная дисциплина программирования
- Предложена Дейкстрой в 1975 г.
- Программа состоит из ограниченного множества математических объектов - структур

# Основные конструкции структурного программирования

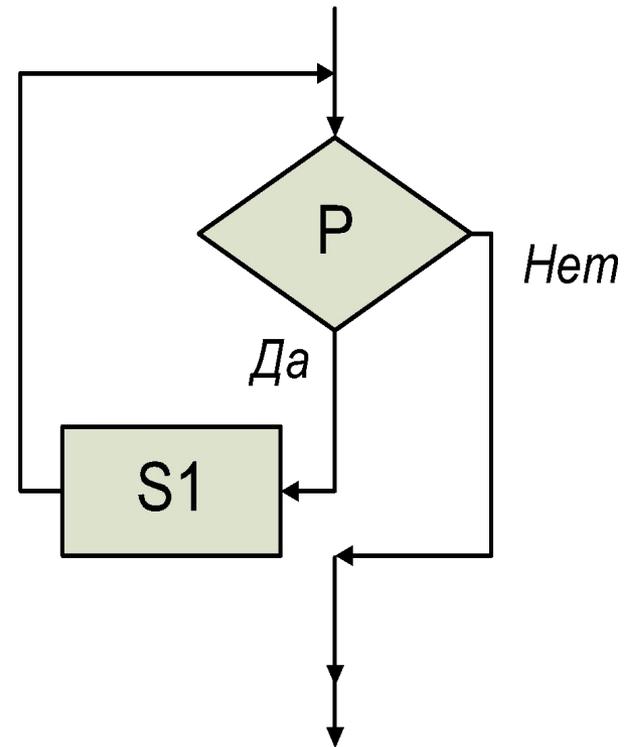
---



Следование



Ветвление



Повторение

# Преимущества структурного программирования

---

- Формализм – все элементы являются математическими объектами
- Для каждой неструктурированной программы можно построить функционально эквивалентную структурную
- Можно доказывать свойства программ

# Свойства программ

---

- Триадны Хоара – запись вида  $\{P\}S\{Q\}$
- Сильные условия
  - $\{x=1\} x := x+3 \{x=4\}$
- Слабые условия
  - $\{x>0\} x := x+3 \{x>3\}$
- Доказанные свойства могут использоваться для автоматизированного синтеза программ

# Расширения структурного программирования

---

- Введены в связи с усложнением программ
- Расширение 1 – переходы из тела цикла
- Расширение 2 – возврат из функции
- Расширение 3 – переход на обработчик исключения

# Понятие псевдокода

---

- ❑ Псевдокод – язык и метод построения алгоритма
- ❑ Введен вместе с технологией пошаговой детализации
- ❑ Позволяет частичную формализацию алгоритма
- ❑ Основан на конструкциях структурного программирования

# Конструкции псевдокода

---

- Следование:
  - Оператор1; Оператор2 ...
- Ветвление
  - ЕСЛИ условие ТО Оператор1  
ИНАЧЕ Оператор2 ВСЕ
- Повторение
  - ПОКА условие ДЕЛАТЬ Оператор ВСЕ

# Пример псевдокода

---

- Уровень 1 :
  - ПОКА список файлов не пуст ДЕЛАТЬ обработать файл;  
ВСЕ
  
- Уровень 2 :
  - ПОКА список файлов не пуст ДЕЛАТЬ открыть файл;  
    ПОКА есть записи в файле ДЕЛАТЬ обработать запись;  
    ВСЕ  
закрыть файл;  
ВСЕ

# Достоинства псевдокода

---

- Частично формализован – может применяться для автоматизации построения кода
- Наглядно представляет алгоритм
- Детализируется до уровня основных операторов выбранного языка

# ООП

---

- Построение систем на основе объектов и их классов
- Включает в себя стадии:
  - Анализа – построения модели предметной области
  - Дизайна – разработки абстракций для выражения модели
  - Программирования – реализации дизайна на языке ООП

# Характеристики ООП

---

- Объекты представляют собой сущности реального мира
- Объекты независимы и содержат информацию о состоянии
- Отсутствует разделение данных – объекты общаются через сообщения
- Объекты могут быть распределенными и функционировать параллельно

# Преимущества ООП

---

- Простота сопровождения программы из-за инкапсуляции поведения в объекте
- Повторное использование объектов
- Очевидное отображение модели реального мира в объектное представление

# Разновидности объектов в ПС

---

- Пассивные - фрагмент информационной среды, который способен хранить разные данные определенного типа (представляющие разные *состояния* этого объекта) и с которым связан некоторый набор операций (*применимых* к этому объекту)
- Активные - расширение пассивного объекта, в котором фрагмент информационной среды способен также хранить и программные фрагменты, способные находиться в процессе выполнения (*в активном состоянии*)

# Особенности ООП

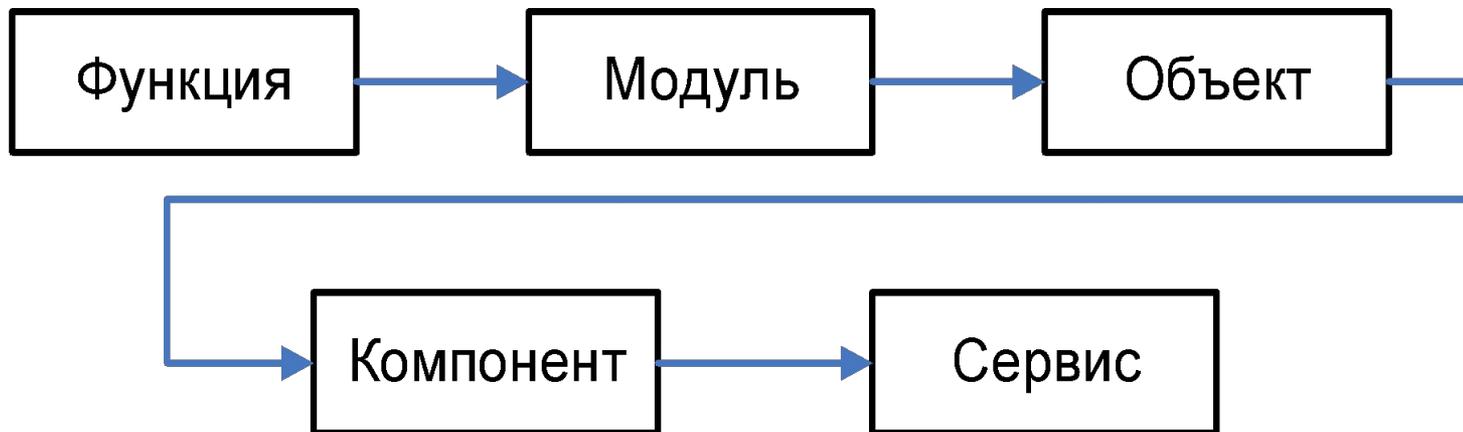
---

- ❑ Определение требований выражается в описании модельного мира
- ❑ Повышается роль прототипирования
- ❑ Расширяется этап внешнего описания
- ❑ Для описания ПС используются различные модели

# Компонентный подход

---

- ❑ Конструирование программ из готовых компонентов
- ❑ Компонент имеет информационную, внешнюю и внутреннюю структуру
- ❑ Модель эволюции компонентов



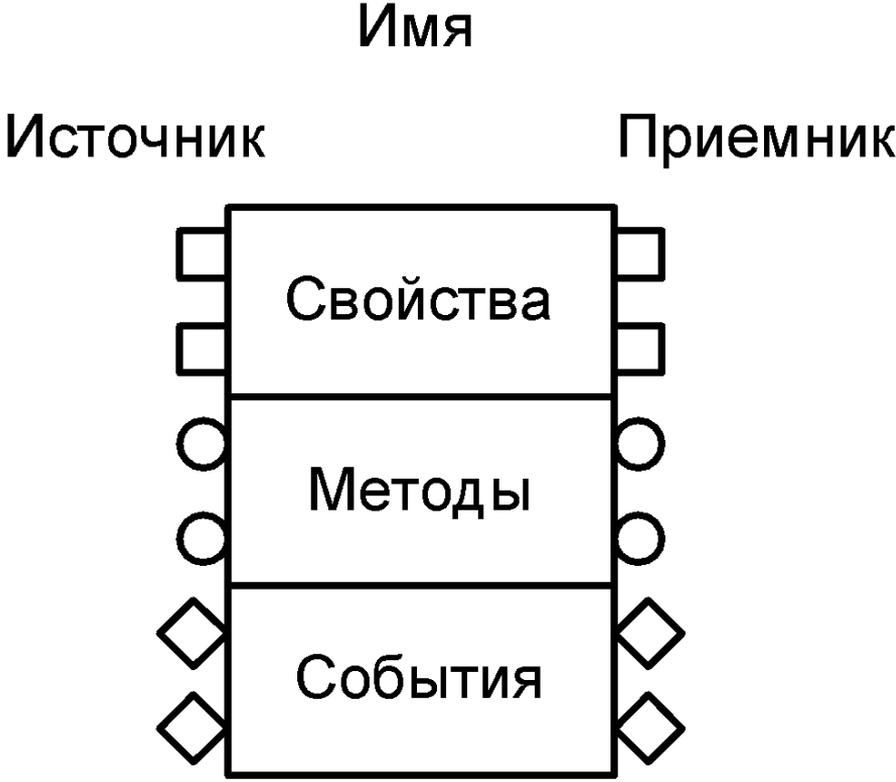
# Отличия ООП и КОП

---

- ❑ КОП основано на интерфейсах, ООП на объектах.
- ❑ КОП – это технология объединения в пакеты и распространения, ООП – технология реализации.
- ❑ КОП предполагает повторное использование очень высокого уровня, ООП – низкого уровня.
- ❑ КОП может быть реализовано на разных языках, ООП ориентировано на отдельный язык.
- ❑ КОП объединяет слабо связанные компоненты, ООП объединяет сильно связанные компоненты, зависящие от отношений между классами.
- ❑ КОП содержит крупнозернистые объекты, ООП – мелкозернистые.
- ❑ КОП поддерживает множество форм динамического связывания и определения сервисов.

# Структура компонента

---



# Элементы компонента

---

- Спецификация компонента – определяет спецификацию программного объекта, описывающую его поведение посредством множества реализуемых интерфейсов.
- Интерфейс компонента определяет множество услуг, предоставляемых компонентом.
- Реализация компонента представляет его воплощение, позволяющее независимо размещать или удалять компонент в структуре системы. Реализация может быть составной, т.е. содержать несколько файлов.
- Инсталляция компонента представляет форму его регистрации в среде времени выполнения.
- Объекты компонента – это экземпляры компонента, создаваемые в процессе работы приложения.

# Инфраструктура компонентов

---

- Модель компонентов – определяет, что является компонентом и как создать новый компонент;
- Модель соединений – определяет способы и средства соединения компонентов в программах;
- Модель развертывания – определяет процесс размещения и интеграции компонентов в среду выполнения.

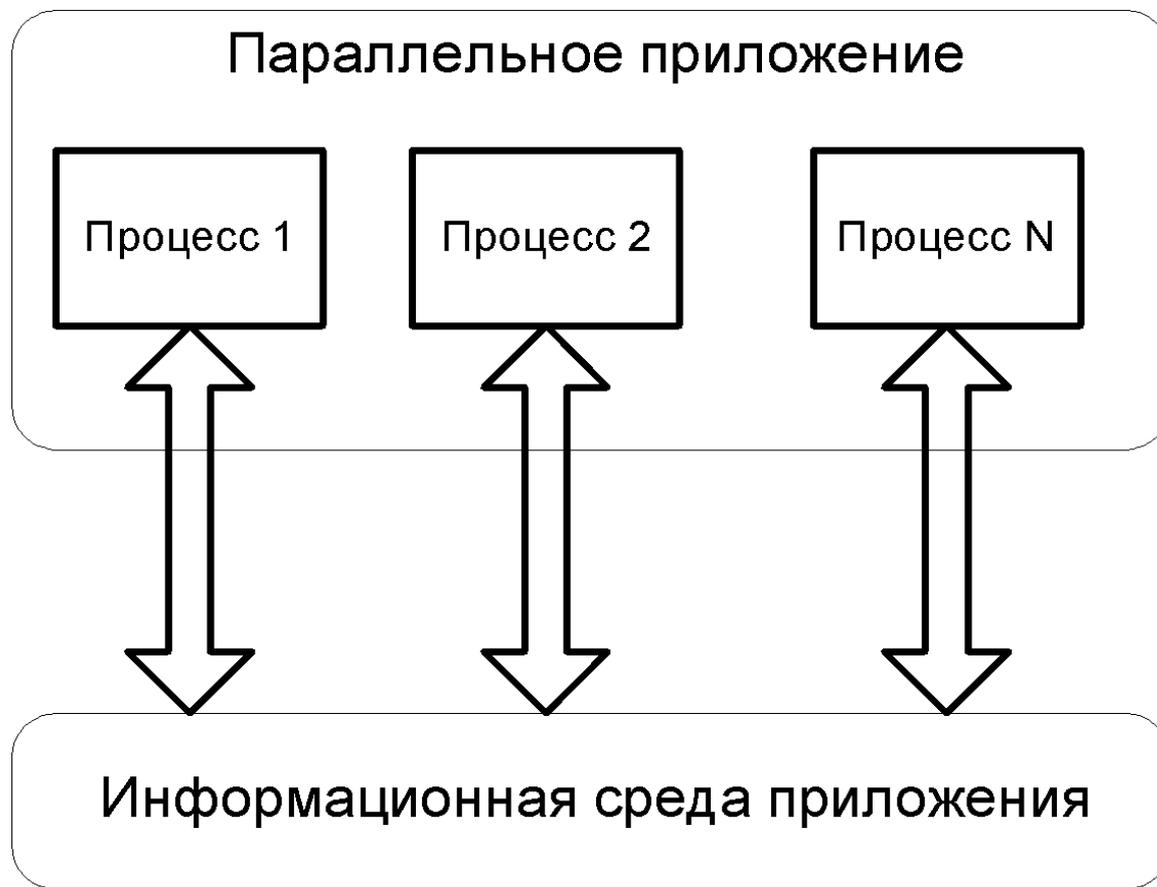
# Параллельное программирование

---

- Процесс решения задач, в которых в один и тот же момент времени могут выполняться одновременно несколько вычислительных операций
- Разработано для решения сложных вычислительных задач
- Ориентировано на высокопроизводительные вычислительные системы

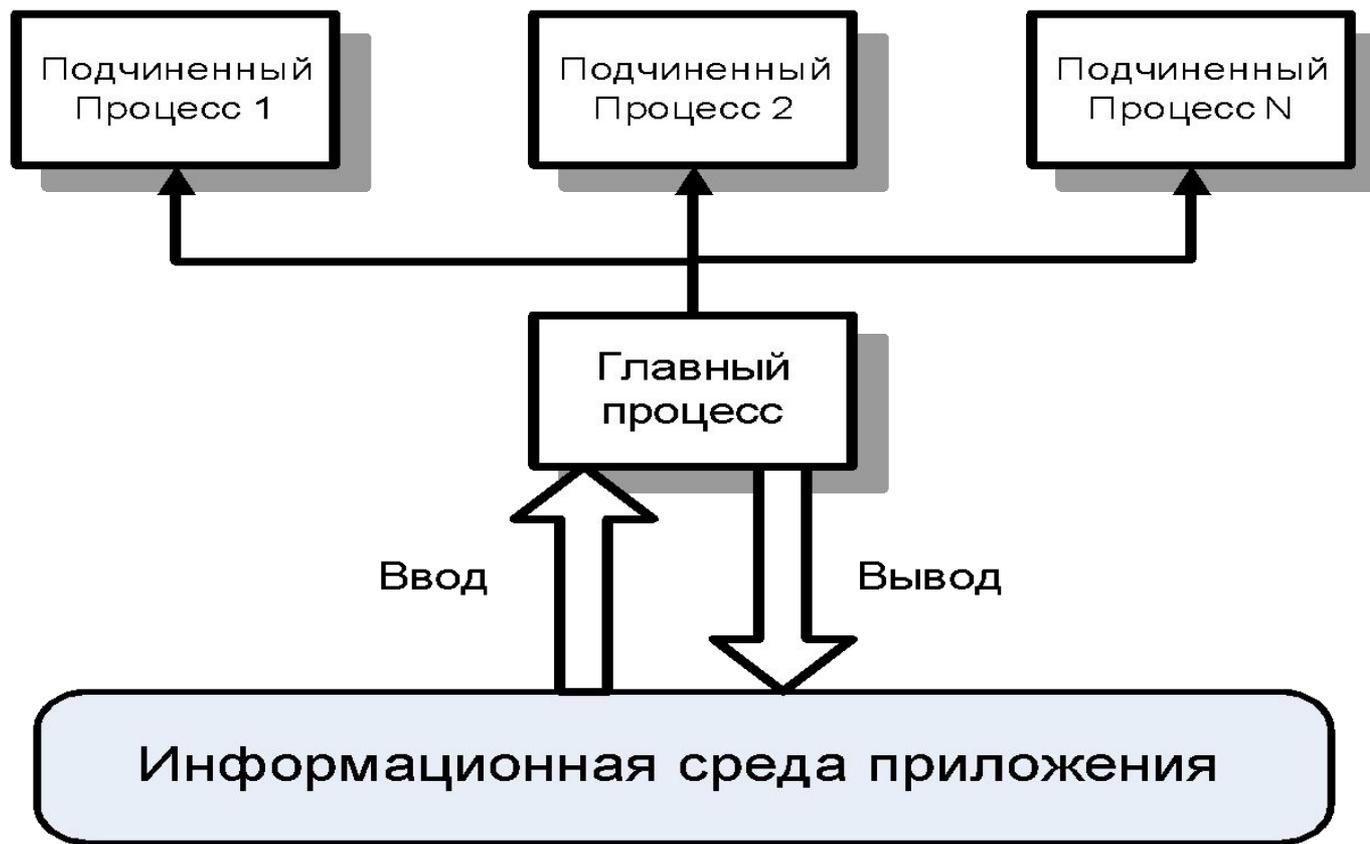
# Модель массового параллелизма

---



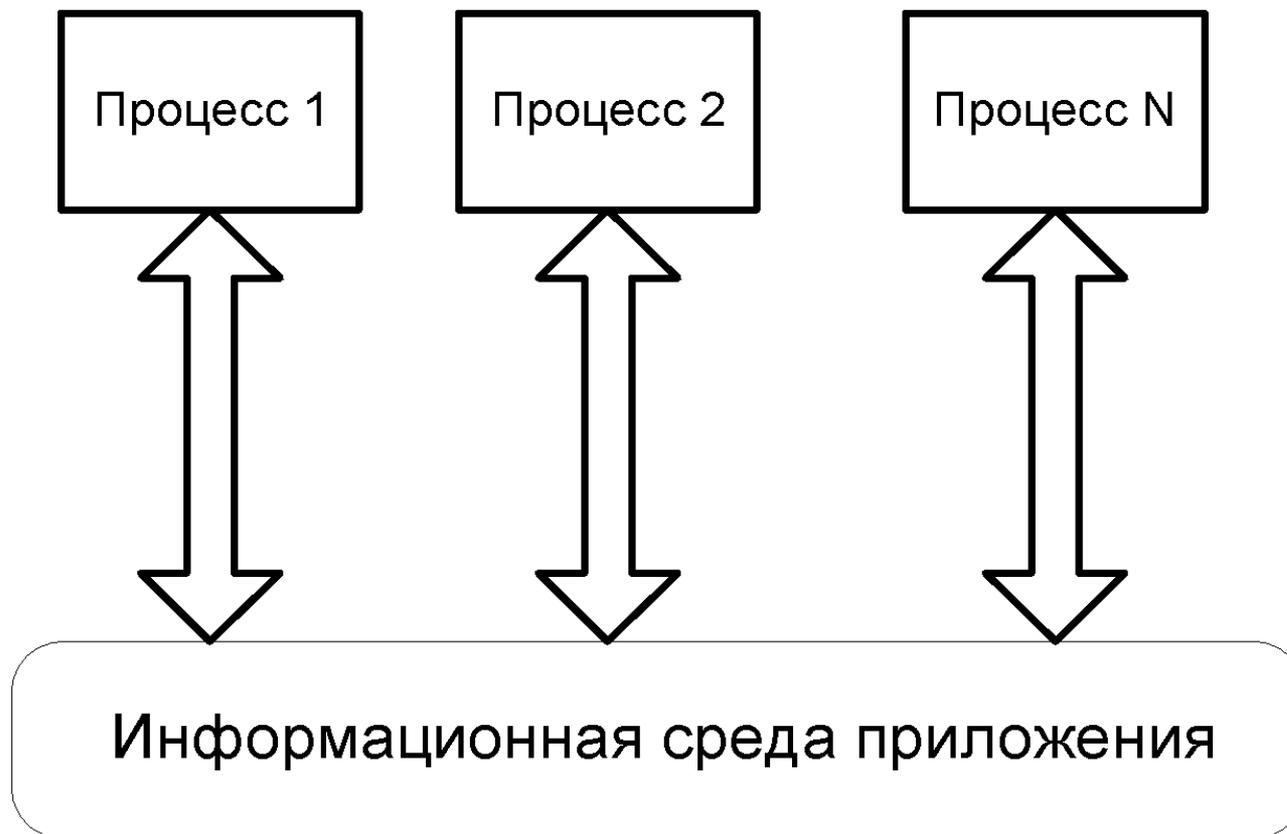
# Модель главный-подчиненный

---



# Модель равноправного взаимодействия

---



# Параллельные платформы

---

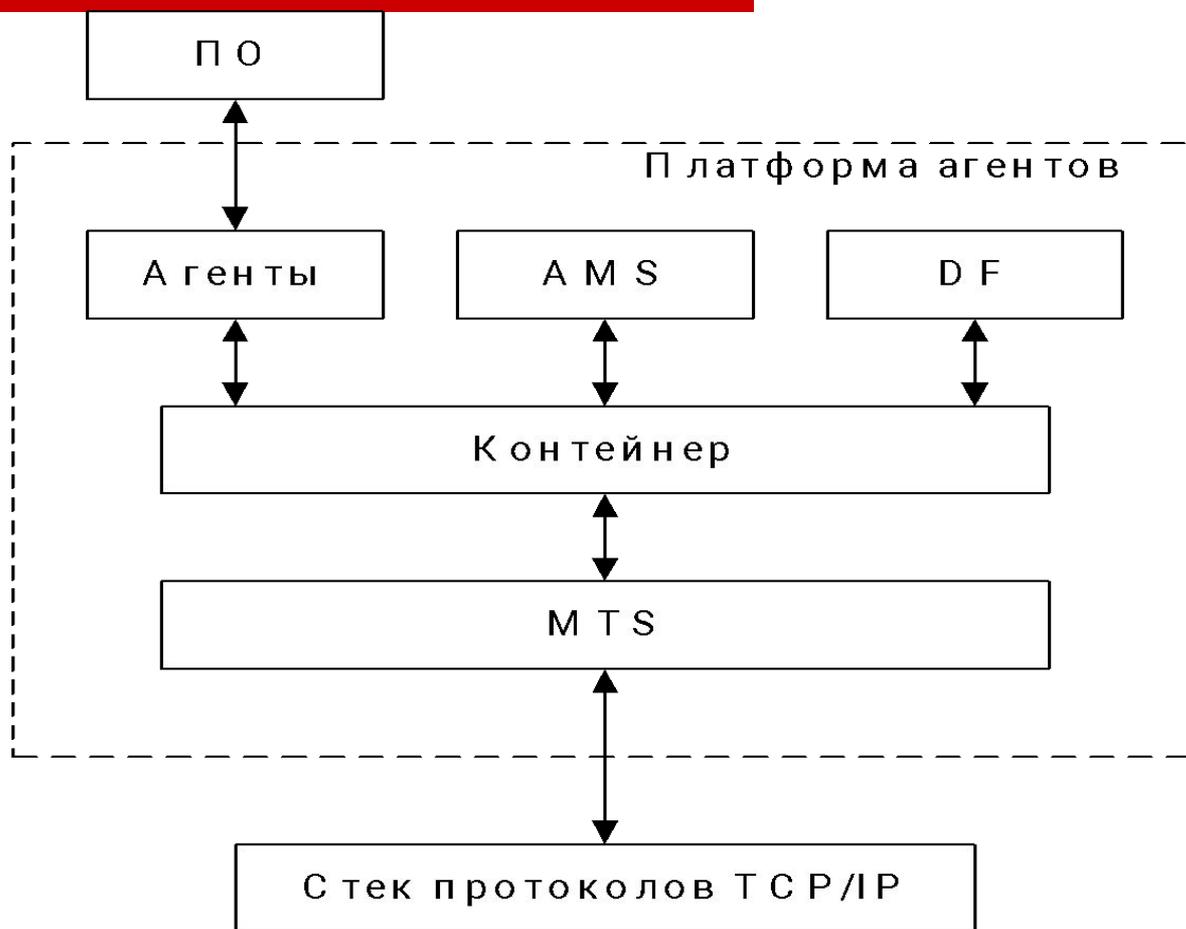
- Определяют интерфейс взаимодействия и механизмы синхронизации процессов
- Реализуются в виде виртуальных машин на разных уровнях ОС
- Примеры: MPI, PVM, OpenMP

# Агентное программирование

---

- Агент – это сущность, которая находится в некоторой среде. От этой среды она получает данные, которые отражают события, происходящие в среде, интерпретирует их и исполняет команды, воздействующие на среду. Агент может содержать программные и аппаратные компоненты.

# Структура агентной платформы



# Структура агента

---

