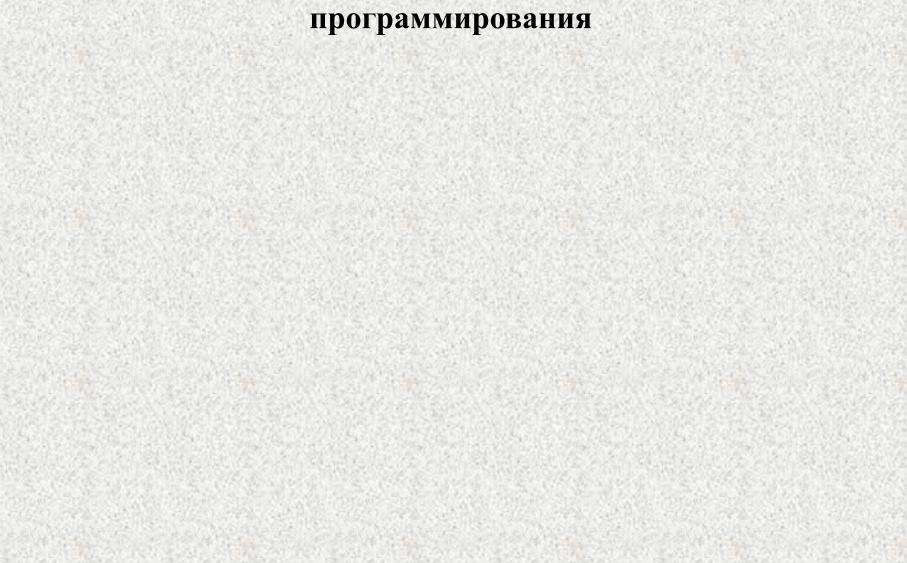
# **Технологии реализации методов современного** программирования



## Этапы решения задач на ЭВМ

Постановка задачи

Формальное построение модели задачи

Построение математической модели задачи

Разработка алгоритма

Программирование

**Анализ результатов** решения задачи

Сопровождение программы

#### 1 Постановка задачи

На этом этапе формулируется цель решения задачи, анализируются требования и подробно описывается содержание задачи, выявляются условия, при которых решается задача, а также определяются входные параметры, которые называются исходными данными

#### 2 Формальное построение модели задачи

На этом этапе составляется формальная модель решения задачи, например, модель базы данных, адекватная оригиналу, модель объектов и потоков информации

#### 3 Построение математической модели решения задачи

На этом этапе составляется формальная модель решения задачи, например, модель базы данных, адекватная оригиналу, модель объектов и потоков информации

# Пример реализации технологии программирования вычислительных задач

#### Постановка задачи

Разработать программу вычисления суммы квадратов целых положительных чисел от 1 до N

#### Формальное построение модели задачи

Вычислить:

$$S=1+4+9+16+...*N^2$$

#### Построение математической модели решения задачи

- 1. Исходные данные: *S*, N
- 2. Математические формулы:

$$s = \sum_{i=1}^{N} i^2.$$

3. Результат S

Пример реализации технологии программирования вычислительных задач

#### Постановка задачи

Разработать программу вычисления суммы квадратов целых положительных чисел от 1 до N

#### Формальное построение модели задачи

Вычислить:

$$S=1+4+9+16+...*N^2$$

# Построение математической модели решения задачи

- 1. Исходные данные: N
- 2. Математические формулы:

$$s = \sum_{i=1}^{N} i^2.$$

- 1. Ввод с клавиатуры **N** целого типа
- 2. Вычисление **S=0, i=1** повторять  $s=s+i^2$ , i=i+1 пока i<=N

3. Результат S

3 Пецэть "°-" €

Пример реализации технологии программирования вычислительных задач

#### Постановка задачи

Разработать программу вычисления суммы квадратов целых положительных чисел от 1 до N

#### Формальное построение модели задачи

Вычислить:

$$S=1+4+9+16+...*N^2$$

### Построение математической модели решения задачи Вычислительные формулы

- Исходные данные: N
- Математические формулы:

$$s = \sum_{i=1}^{N} i^2$$
.  
Результат **S**

- Ввод с клавиатуры **N** целого типа
- Вычисление **S=0**, *i*=1

повторять s=s+i, t=i+1 пока i <= 1

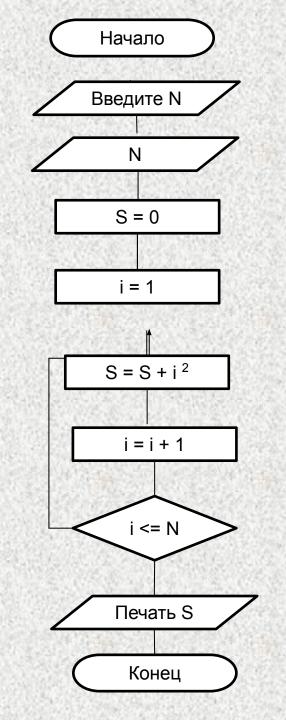
Представление математической модели решения задачи в виде вычислительных формул:

- 1. Ввод с клавиатуры **N** целого типа
- 2. Вычисление **S=0, i=1**

повторять  $s=s+i^2$ , t=i+1 пока i <=N

3. Печать "S=", **S** 

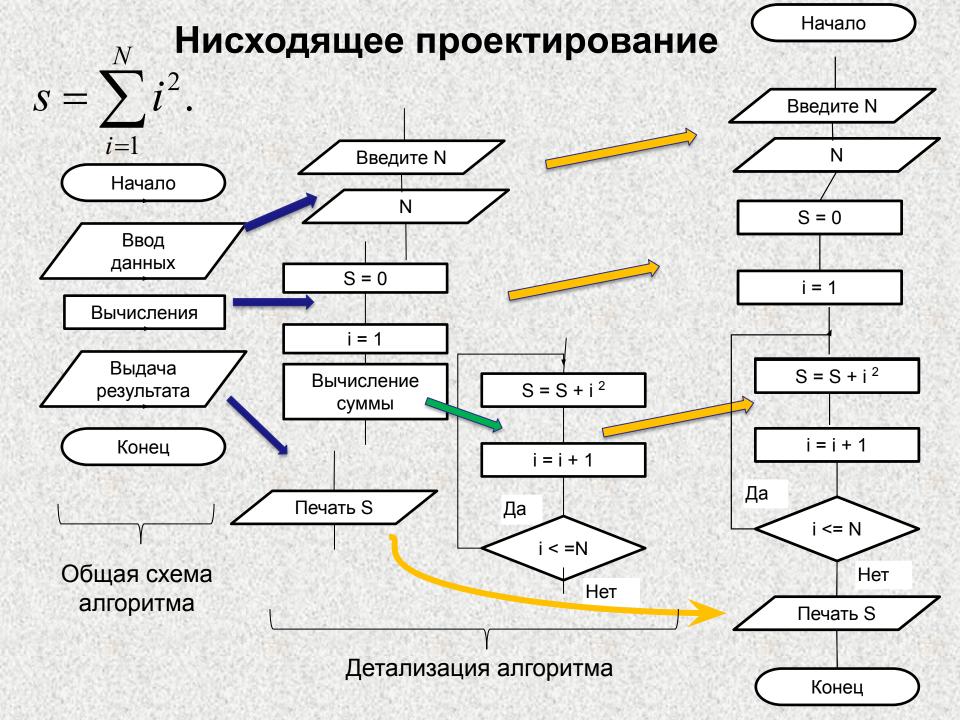
позволяет записать алгоритм, представив последовательность выполнения вычислений в виде последовательности базовых структур



# **ЗТехнологии реализации методов** разработки алгоритмов и программ

Нисходящее проектирование — технология разработки программ, при которой на каждом шаге проектирования задача разбивается на более мелкие подзадачи так, что в любой момент разработки имеется действующий вариант программы в терминах выделенных подзадач.

Восходящее проектирование — технология разработки программ, при которой сначала проектируются и отлаживаются подпрограммы для выполнения простых операций, после чего они связываются в единую программу.



### Основные достоинства нисходящего проектирования:

- проявление логики программы возникает уже при чтении головного модуля,
  что делает программу боле простой;
- возможность контроля хода работы над программой в процессе последовательной детализации программы обеспечивает ее непрерывную корректировку; отсутствие комплексной отладки благодаря сквозному контролю позволяет сэкономить до 30 % общего времени разработки программ;
- одновременная параллельная работа нескольких программистов может оказаться эффективной.

При нисходящем проектировании, однако, возможны и такие ситуации, когда после значительных затрат на программирование выясняется необходимость объединения нескольких подзадач в один модуль, либо обнаруживается невозможность выполнения модулями нижних уровней своих функций при заданных временных ограничениях.

# Восходящее проектирование

Восходящее проектирование (или проектирование «снизу вверх») основано на выделении нескольких достаточно крупных модулей, реализующих некоторые функции в общей программе.

При выделении модулей опираются на доступность реализуемых функций для понимания, простоту структурирования данных, существование готовых программ и модулей для реализации заданных функций, возможности переделки существующих программ для новых целей; имеет значение и размер будущего модуля.

Каждый модуль при восходящем проектировании автономно программируется, тестируется и отлаживается.

После этого отдельные модули объединяются в подсистемы с помощью управляющего модуля, в котором определяется последовательность вызовов модулей, ввод-вывод и контроль данных и результатов.

В свою очередь, подсистемы затем объединяются в более сложные системы и в общий программный комплекс, который подвергается комплексной отладке с проверкой правильности межмодульных связей.