

# ЭВОЛЮЦИЯ ЯЗЫКОВ

Псевдокоды. Функциональные языки. Скриптовые языки.



**СГУГиТ**  
СИБИРСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ГЕОСИСТЕМ И ТЕХНОЛОГИЙ

# СОДЕРЖАНИЕ ЛЕКЦИИ

- ☐ Классификация языков
- ☐ Машинные языки
- ☐ Языки ассемблера
- ☐ Языки высокого уровня
- ☐ Парадигмы



# КЛАССИФИКАЦИЯ ЯЗЫКОВ

Для создания программ люди используют языки, похожие на человеческий язык. Результаты переводятся в машинный код, который понимают компьютеры.

Языки программирования делятся на три большие категории:

- Машинные языки
- Языки ассемблера
- Языки высокого уровня



**СГУГиТ**

СИБИРСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ГЕОСИСТЕМ И ТЕХНОЛОГИЙ

# МАШИННЫЕ ЯЗЫКИ

Машинные языки (языки первого поколения) являются наиболее основным типом компьютерных языков, состоящих из строк чисел, которые может использовать компьютерное оборудование.

Различные типы оборудования используют разный машинный код. Например, компьютеры IBM используют другой машинный язык, чем компьютеры Apple.



# ОПРЕДЕЛЕНИЕ

**Язык ассемблера** – это низкоуровневый язык программирования, который используется для написания программ, работающих непосредственно на аппаратном уровне компьютера. Он представляет собой набор инструкций, которые выполняются процессором компьютера. Каждая инструкция ассемблера соответствует определенной операции, которую процессор может выполнить, такой как сложение чисел или перемещение данных из одной области памяти в другую.



**СГУГиТ**

СИБИРСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ГЕОСИСТЕМ И ТЕХНОЛОГИЙ

# ЯЗЫКИ АССЕМБЛЕРА

Языки ассемблера (языки второго поколения) лишь несколько проще в работе, чем с машинными языками.

Для создания программ на языке ассемблера разработчики используют загадочные английские фразы для представления строк чисел.

Затем код преобразуется в объектный код с помощью переводчика, называемого ассемблером.



# ЯЗЫКИ АССЕМБЛЕРА

```
// I = 15;  
MOV R3, #15  
STR R3, [R11, #-8]
```

```
// J = 25;  
MOV R3, #25  
STR R3, [R11, #-12]
```

```
// I = I + J;  
LDR R2, [R11, #-8]  
LDR R3, [R11, #-12]  
ADD R3, R2, R3  
STR R3, [R11, #-8]
```

ASSEMBLY LANGUAGE

ASSEMBLER

```
1100 1010 1011 0011  
1100 1010 1011 0011  
1100 1010 1011 0011  
1100 1010 1011 0011  
1100 1010 1011 0011  
1100 1010 1011 0011
```

MACHINE CODE



СГУГиТ

СИБИРСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ГЕОСИСТЕМ И ТЕХНОЛОГИЙ

# ЯЗЫКИ ВЫСОКОГО УРОВНЯ

Языки более высокого уровня более мощны, чем язык ассемблера, и позволяют программисту работать в более английской среде.

Языки программирования более высокого уровня разделены на три "поколения", каждое из которых более мощное, чем предыдущее:

- Языки третьего поколения
- Языки четвертого поколения
- Языки пятого поколения





# ТРЕТЬЕ ПОКОЛЕНИЕ

Языки третьего поколения первыми используют настоящие английские фразы, что делает их проще в использовании, чем предыдущие языки.

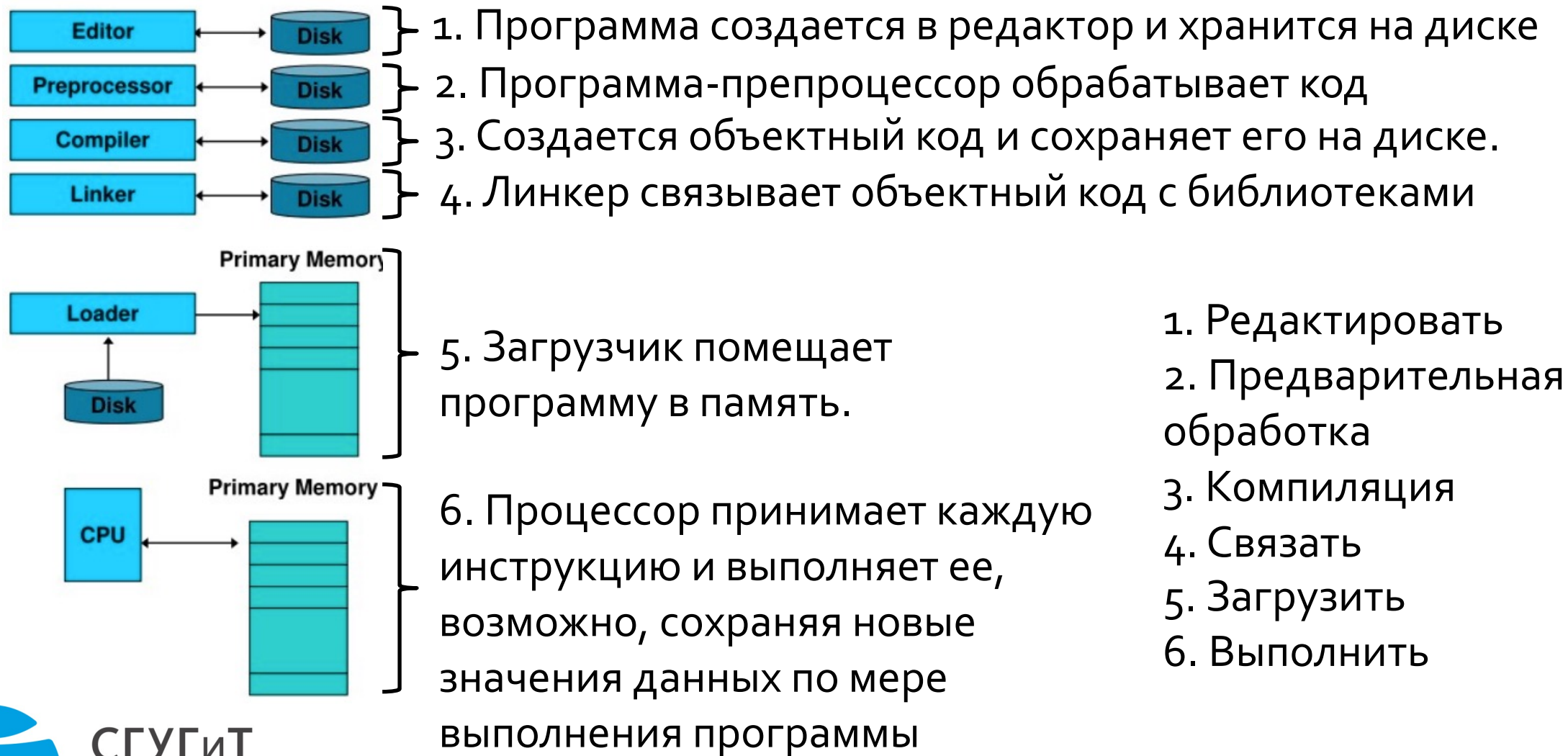
Языки третьего поколения являются портативными, что означает, что объектный код, созданный для одного типа системы, может быть переведен для использования в другом типе системы.

Следующие языки являются языками третьего поколения:

- FORTRAN
- COBOL
- BASIC
- Pascal
- C/C++
- ALGOL
- Java
- ActiveX



# ФАЗЫ ПРОГРАММ



# FORTRAN

```
! Fortran 95 Example program
! Input:  An integer, List_Len, where List_Len is less
!         than 100, followed by List_Len-Integer values
! Output: The number of input values that are greater
!         than the average of all input values
Implicit none
Integer Dimension(99) :: Int_List
Integer :: List_Len, Counter, Sum, Average, Result
Result= 0
Sum = 0
Read *, List_Len
If ((List_Len > 0) .AND. (List_Len < 100)) Then
! Read input data into an array and compute its sum
  Do Counter = 1, List_Len
    Read *, Int_List(Counter)
    Sum = Sum + Int_List(Counter)
  End Do
! Compute the average
  Average = Sum / List_Len
! Count the values that are greater than the average
  Do Counter = 1, List_Len
    If (Int_List(Counter) > Average) Then
      Result = Result + 1
    End If
  End Do
! Print the result
  Print *, 'Number of values > Average is:', Result
Else
  Print *, 'Error - list length value is not legal'
End If
End Program Example
```



**СГУГиТ**

СИБИРСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ГЕОСИСТЕМ И ТЕХНОЛОГИЙ

# ALGOL 6o

```
comment ALGOL 60 Example Program
Input:  An integer, listlen, where listlen is less than
        100, followed by listlen-integer values
Output: The number of input values that are greater than
        the average of all the input values  ;

begin
  integer array intlist [1:99];
  integer listlen, counter, sum, average, result;
  sum := 0;
  result := 0;
  readint (listlen);
  if (listlen > 0)  $\wedge$  (listlen < 100) then
    begin
comment Read input into an array and compute the average;
      for counter := 1 step 1 until listlen do
        begin
          readint (intlist[counter]);
          sum := sum + intlist[counter]
        end;
comment Compute the average;
        average := sum / listlen;
comment Count the input values that are > average;
        for counter := 1 step 1 until listlen do
          if intlist[counter] > average
            then result := result + 1;
comment Print result;
        printstring("The number of values > average is:");
        printint (result)
      end
    else
      printstring ("Error-input list length is not legal";
    end
  end
```



**СГУГиТ**

СИБИРСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ГЕОСИСТЕМ И ТЕХНОЛОГИЙ

# BASIC

```
REM  BASIC Example Program
REM  Input:  An integer, listlen, where listlen is less
REM          than 100, followed by listlen-integer values
REM  Output: The number of input values that are greater
REM          than the average of all input values
      DIM intlist(99)
      result = 0
      sum = 0
      INPUT listlen
      IF listlen > 0 AND listlen < 100 THEN
REM  Read input into an array and compute the sum
        FOR counter = 1 TO listlen
          INPUT intlist(counter)
          sum = sum + intlist(counter)
        NEXT counter
REM  Compute the average
        average = sum / listlen
REM  Count the number of input values that are > average
        FOR counter = 1 TO listlen
          IF intlist(counter) > average
            THEN result = result + 1
          NEXT counter
REM  Print the result
        PRINT "The number of values that are > average is:";
          result
      ELSE
        PRINT "Error-input list length is not legal"
      END IF
    END
```



**СГУГиТ**

СИБИРСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ГЕОСИСТЕМ И ТЕХНОЛОГИЙ

# ЧЕТВЕРТОЕ ПОКОЛЕНИЕ

Языки четвертого поколения еще проще в использовании, чем языки третьего. поколения.

Языки четвертого поколения могут использовать текстовую среду или могут позволить программисту работать в визуальной среде с использованием графических инструментов.

Следующие языки являются языками четвертого поколения:

- Visual Basic (VB)
- Python
- Perl
- Ruby
- C#



# JAVA

```
REM  BASIC Example Program
REM  Input:  An integer, listlen, where listlen is less
REM          than 100, followed by listlen-integer values
REM  Output: The number of input values that are greater
REM          than the average of all input values
      DIM intlist(99)
      result = 0
      sum = 0
      INPUT listlen
      IF listlen > 0 AND listlen < 100 THEN
REM  Read input into an array and compute the sum
        FOR counter = 1 TO listlen
          INPUT intlist(counter)
          sum = sum + intlist(counter)
        NEXT counter
REM  Compute the average
        average = sum / listlen
REM  Count the number of input values that are > average
        FOR counter = 1 TO listlen
          IF intlist(counter) > average
            THEN result = result + 1
          NEXT counter
REM  Print the result
        PRINT "The number of values that are > average is:";
          result
      ELSE
        PRINT "Error-input list length is not legal"
      END IF
    END
```



**СГУГиТ**

СИБИРСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ГЕОСИСТЕМ И ТЕХНОЛОГИЙ

# Perl

```
# Perl Example Program
# Input:  An integer, $listlen, where $listlen is less
#         than 100, followed by $listlen-integer values.
# Output: The number of input values that are greater than
#         the average of all input values.
($sum, $result) = (0, 0);
$listlen = <STDIN>;
if (($listlen > 0) && ($listlen < 100)) {
# Read input into an array and compute the sum
  for ($counter = 0; $counter < $listlen; $counter++) {
    $intlist[$counter] = <STDIN>;
  } #- end of for (counter ...)
# Compute the average
  $average = $sum / $listlen;
# Count the input values that are > average
  foreach $num (@intlist) {
    if ($num > $average) { $result++; }
  } #- end of foreach $num ...
# Print result
  print "Number of values > average is: $result \n";
} #- end of if (($listlen ...
else {
  print "Error--input list length is not legal \n";
}
```



**СГУГиТ**

СИБИРСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ГЕОСИСТЕМ И ТЕХНОЛОГИЙ



# C#

```
// C# Example Program
// Input:  An integer, listlen, where listlen is less than
//         100, followed by listlen-integer values.
// Output: The number of input values that are greater
//         than the average of all input values.
using System;
public class Ch2example {
    static void Main() {
        int[] intlist;
        int listlen,
            counter,
            sum = 0,
            average,
            result = 0;
        intList = new int[99];
        listlen = Int32.Parse(Console.ReadLine());
        if ((listlen > 0) && (listlen < 100)) {
// Read input into an array and compute the sum
            for (counter = 0; counter < listlen; counter++) {
                intList[counter] =
                    Int32.Parse(Console.ReadLine());
                sum += intList[counter];
            } //- end of for (counter ...)
// Compute the average
            average = sum / listlen;
// Count the input values that are > average
            foreach (int num in intList)
                if (num > average) result++;
// Print result
            Console.WriteLine(
                "Number of values > average is:" + result);
        } //- end of if ((listlen ...
        else
            Console.WriteLine(
                "Error--input list length is not legal");
        } //- end of method Main
    } //- end of class Ch2example
}
```



**СГУГиТ**

СИБИРСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ГЕОСИСТЕМ И ТЕХНОЛОГИЙ

# ПЯТОЕ ПОКОЛЕНИЕ

Языки пятого поколения являются предметом дебатов в сообществе программирования - некоторые программисты не могут согласиться с тем, что они даже существуют.

Эти языки высокого уровня будут использовать искусственный интеллект для создания программ.



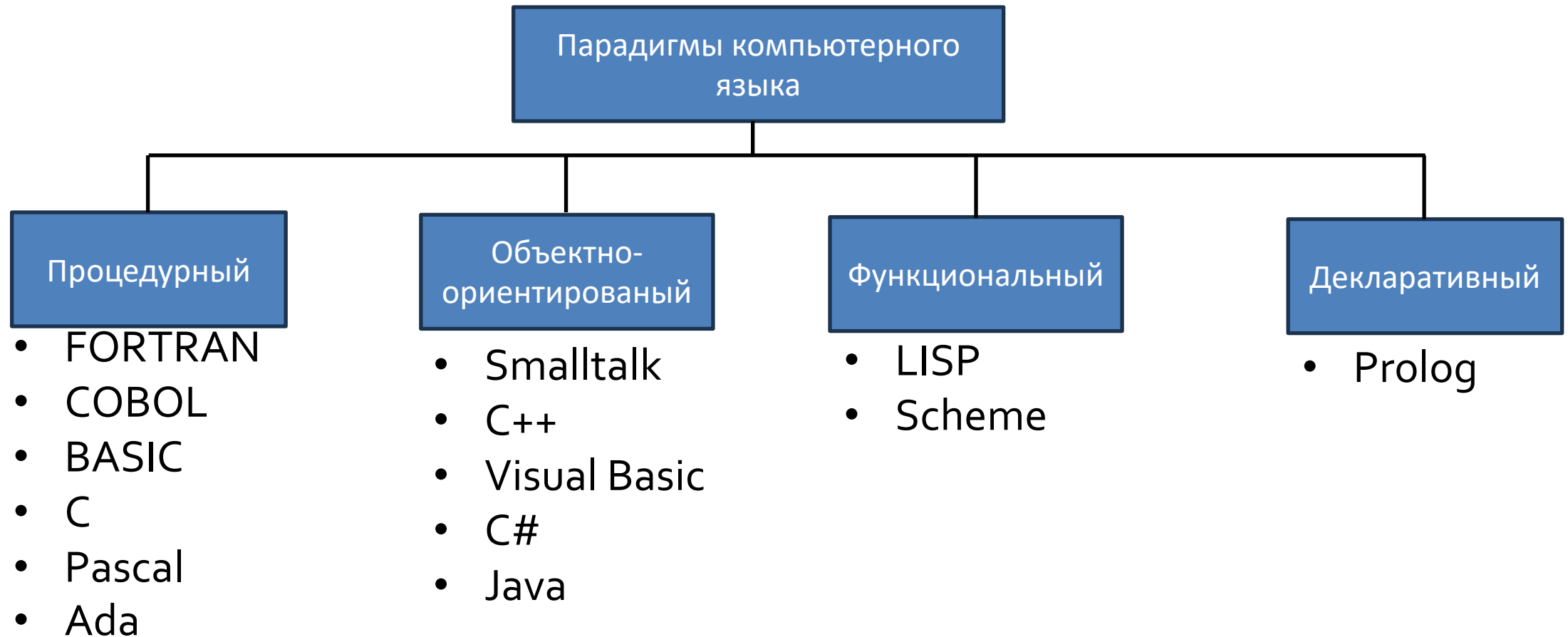
# ПАРАДИГМЫ

Сегодня компьютерные языки делятся на категории в зависимости от того, какой подход они используют для решения той или иной задачи.

Парадигма, таким образом - это способ, с помощью которого компьютерный язык смотрит на решаемую проблему. Мы делим компьютерные языки на четыре парадигмы: процедурную, объектно-ориентированную, функциональную и декларативную.



# ПАРАДИГМЫ



СГУГиТ

СИБИРСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ГЕОСИСТЕМ И ТЕХНОЛОГИЙ

«**Язык программирования** – это инструмент, который позволяет нам выразить наши идеи о том, как решать проблемы, и научить машину исполнять эти идеи.»

Линус Торвальдс



**СГУГиТ**  
СИБИРСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ГЕОСИСТЕМ И ТЕХНОЛОГИЙ

Гришин Руслан Владимирович  
mail to: [r.grishin54@gmail.com](mailto:r.grishin54@gmail.com)