

Севастопольский государственный университет
Кафедра информационных систем

Курс лекций по дисциплине
«Алгоритмизация и программирование»

Лектор: Бондарев Владимир Николаевич

Лекция 2

**Состав и назначение программного обеспечения.
Классификация языков программирования. Схема
развития алгоритмических языков.**

Состав и назначение программного обеспечения (ПО)

Под **программным обеспечением** (Software) понимается совокупность программ, выполняемых вычислительной системой.

Программное обеспечение ЭВМ состоит из трех основных компонентов:

- **системного ПО**, представляющего набор программ, обеспечивающих порядок прохождения задач через ЭВМ, распределение и управление ресурсами ЭВМ и взаимодействие с пользователем;
- **инструментальных программных систем (систем программирования)**, облегчающих процесс создания новых программ для компьютера;
- **прикладных программ**, направленных на решение задач пользователя.

Прикладная программа — это любая конкретная программа, способствующая решению какой-либо задачи в пределах данной предметной области.

Операционная система (ОС)

Операционная система — это комплекс взаимосвязанных системных программ, назначение которого — организовать взаимодействие пользователя с компьютером и выполнение всех других программ.

ОС обычно хранится во внешней памяти компьютера. При включении компьютера она считывается из памяти и размещается в ОЗУ. Этот процесс называется **загрузкой операционной системы**.

В функции ОС входит:

- осуществление диалога с пользователем;
- ввод-вывод и управление данными;
- планирование и организация процесса обработки программ;
- распределение ресурсов (оперативной памяти и кэша, процессора, внешних устройств);
- запуск программ на выполнение;
- всевозможные вспомогательные операции обслуживания;
- передача информации между различными внутренними устройствами;
- программная поддержка работы периферийных устройств (дисплея, клавиатуры, дисковых накопителей, принтера и др.).

Пакеты прикладных программ

Пакеты прикладных программ (ППП) — это специальным образом организованные программные комплексы, рассчитанные на общее применение в определенной проблемной области и дополненные соответствующей технической документацией.

В зависимости от характера решаемых задач различают следующие разновидности ППП:

- пакеты для решения типовых инженерных, планово-экономических, общенаучных задач;
- пакеты системных программ;
- пакеты для обеспечения систем автоматизированного проектирования и систем автоматизации научных исследований и др.

Интегрированные пакеты представляют собой набор нескольких программных продуктов, объединенных в единый удобный инструмент. Наиболее развитые из них включают в себя текстовый редактор, органайзер, электронную таблицу, СУБД, средства поддержки электронной почты, программу создания презентационной графики.

Система программирования

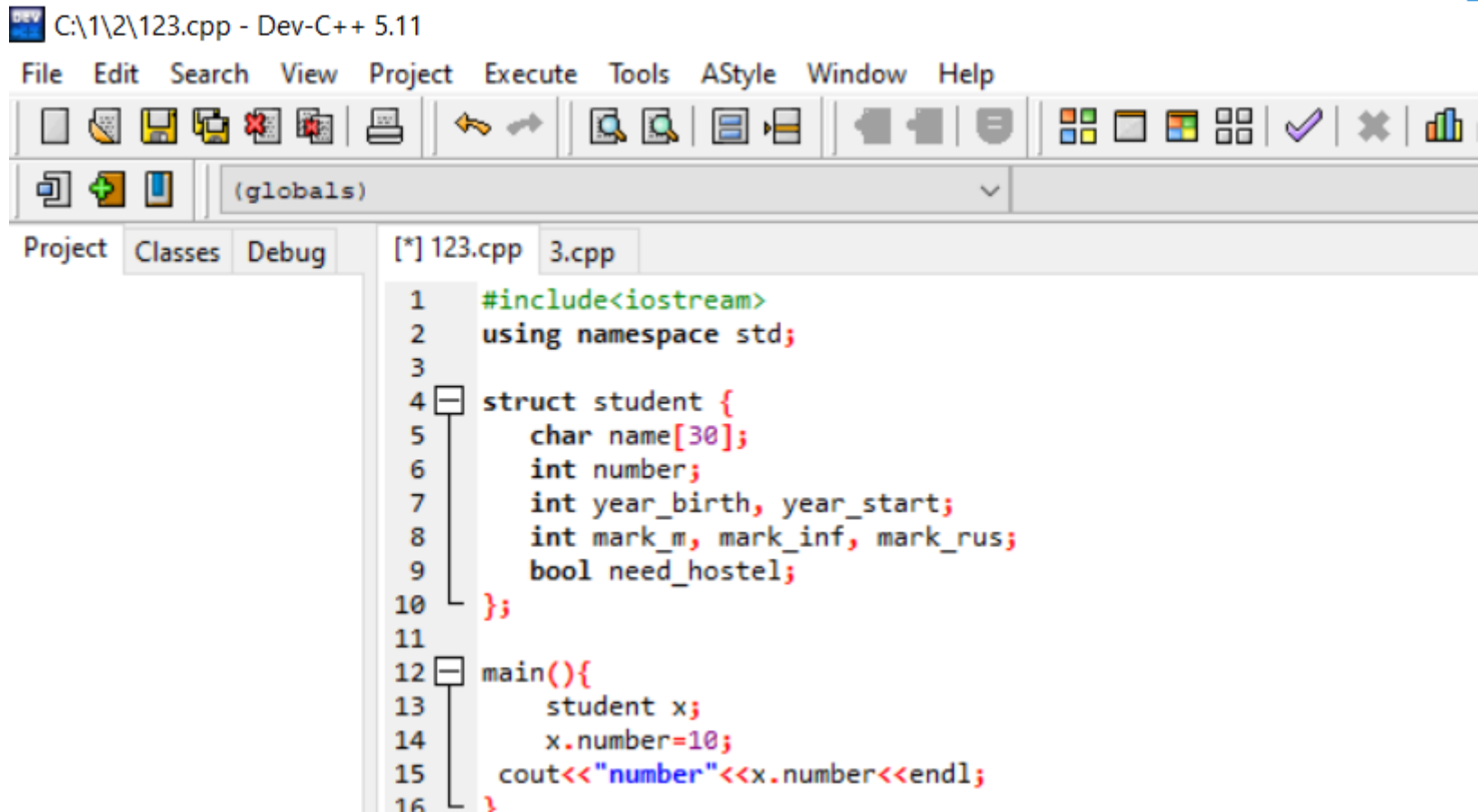
Система программирования — это система для разработки новых программ на конкретном языке программирования.

Современные системы программирования обычно предоставляют пользователям **мощные и удобные средства разработки программ**. В них входят:

- компилятор или интерпретатор;
- средства создания и редактирования текстов программ;
- библиотеки стандартных программ и функций;
- отладчики т.е. программы, помогающие находить и устранять ошибки в программе;
- "дружественная" к пользователю диалоговая среда;
- графические библиотеки;
- встроенная справочная служба.

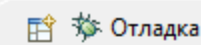
В курсе АиП используются следующие системы программирования: **Dev-C++** или **Eclipse CDT**

Система программирования DevC++



The screenshot displays the Dev-C++ 5.11 interface. The title bar indicates the file path 'C:\1\2\123.cpp - Dev-C++ 5.11'. The menu bar includes 'File', 'Edit', 'Search', 'View', 'Project', 'Execute', 'Tools', 'AStyle', 'Window', and 'Help'. The toolbar contains icons for file operations (new, open, save, print), editing (undo, redo), and execution (run, step-through). The 'Project' panel on the left shows a tree view with 'globals', '123.cpp', and '3.cpp'. The main editor window displays the following C++ code:

```
1  #include<iostream>
2  using namespace std;
3
4  struct student {
5      char name[30];
6      int number;
7      int year_birth, year_start;
8      int mark_m, mark_inf, mark_rus;
9      bool need_hostel;
10 };
11
12 main(){
13     student x;
14     x.number=10;
15     cout<<"number"<<x.number<<endl;
16 }
```



Структура проекта

- lab1
- oop1
- SearchIOS
- task1_2
 - Binaries
 - Includes
 - Debug
 - main.cpp
 - point.cpp
 - point.h
 - triangle.cpp
 - triangle.h
- zadacha1_1
- zadacha1_2
- zadacha1_2_v2

main.cpp

```

1/*
2 * main.cpp
3 *
4 * Created on: 18.05.2011
5 * Author: admin
6 */
7#include <iostream>
8#include "Triangle.h"
9//#include "CyrIOS.h" // for Visual C++ 6.0
10
11
12using namespace std;
13
14int Menu();
15int GetNumber(int, int);
16void ExitBack();
17void Show(Triangle* [], int);
18void Move(Triangle* [], int);
19void FindMax(Triangle* [], int);
20void IsIncluded(Triangle* [], int);
21double GetDouble();

```

Схема

- iostream
- Triangle.h
- std
- Menu() : int
- GetNumber(int, int) : int
- ExitBack() : void
- Show(Triangle*[], int) : void
- Move(Triangle*[], int) : void
- FindMax(Triangle*[], int) : void
- IsIncluded(Triangle*[], int) : void
- GetDouble() : double
- Triangle::count : int
- main() : int
- Menu() : int
- GetNumber(int, int) : int
- ExitBack() : void
- Show(Triangle*[], int) : void
- Move(Triangle*[], int) : void

Неполадки Задачи Консоль Властивости Debug

C-Build [task1_2]

g++ -otask1_2.exe triangle.o point.o main.o

Info: resolving std::cout by linking to __imp__ZSt4cout (auto-import)

c:/mingw/bin/./lib/gcc/mingw32/4.5.0/./../../../../mingw32/bin/ld.exe: warning:

auto-importing has been activated without --enable-auto-import specified on the command line.

Info: resolving std::cin by linking to __imp__ZSt3cin (auto-import)

This should work unless it involves constant data structures referencing symbols from auto-imported DLLs.

Build complete for project task1_2

ТЕХНОЛОГИЯ СОЗДАНИЯ ПРОГРАММ

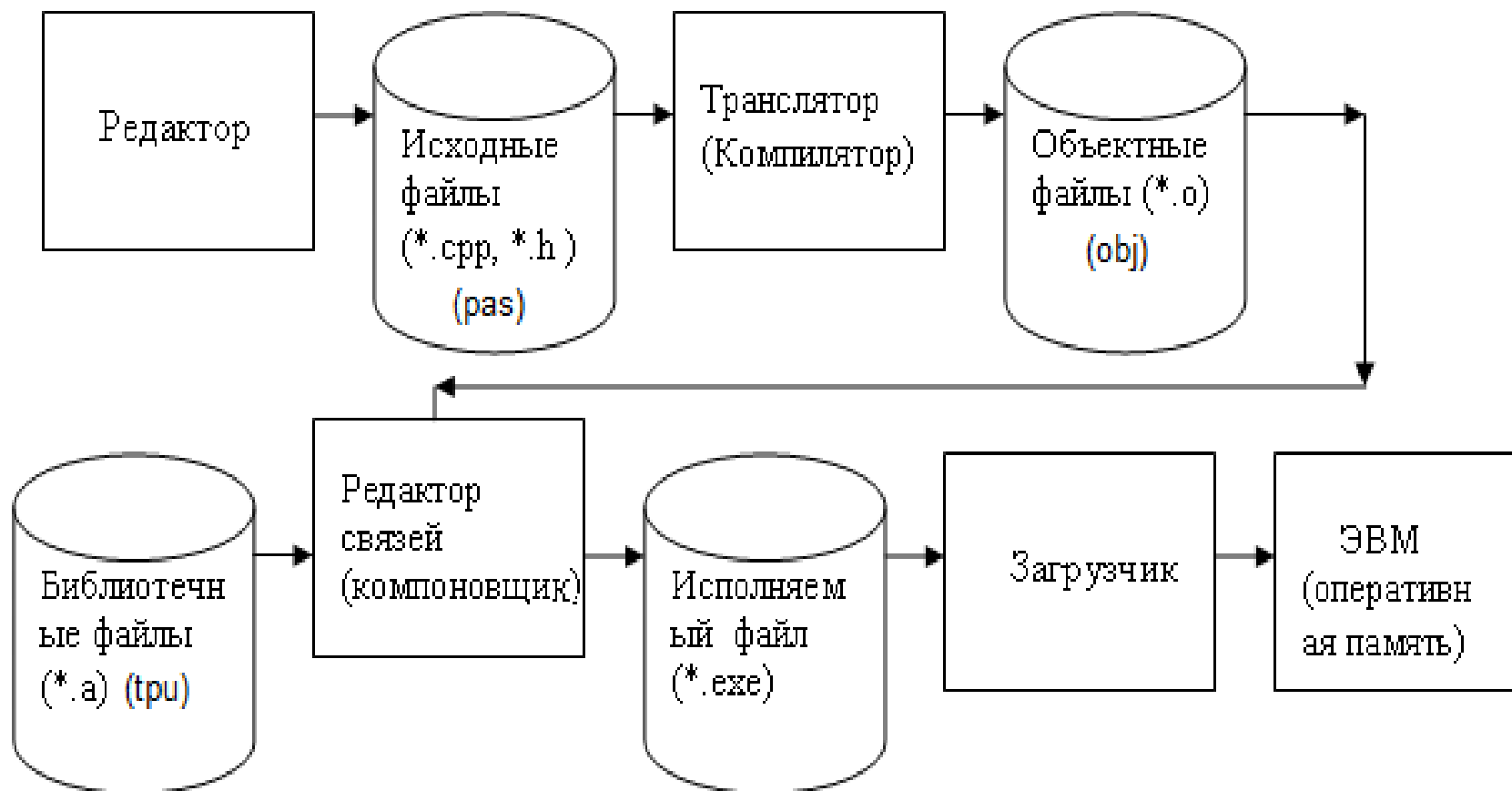
Этапы разработки программы:

- 1) постановка задачи;
- 2) проектирование программы (разработка алгоритма);
- 3) кодирование;
- 4) отладка программы
 - ошибки трансляции (compiler error);
 - ошибки времени выполнения (run-time error);
 - семантические ошибки;
 - ошибки входных данных;
- 5) эксплуатация программы
 - (Руководство разработчика программы, Руководство пользователя).

Отладка это процесс многократного выполнения программы с разными вариантами входных данных. Целенаправленная проверка работоспособности программы называется *тестированием*.

ТЕХНОЛОГИЯ СОЗДАНИЯ ПРОГРАММ

Преобразование программ системой программирования



Язык программирования и его составляющие

Язык программирования (programming language) - формализованный язык, предназначенный для описания процессов решения задач на ЭВМ. Языки программирования являются искусственными.

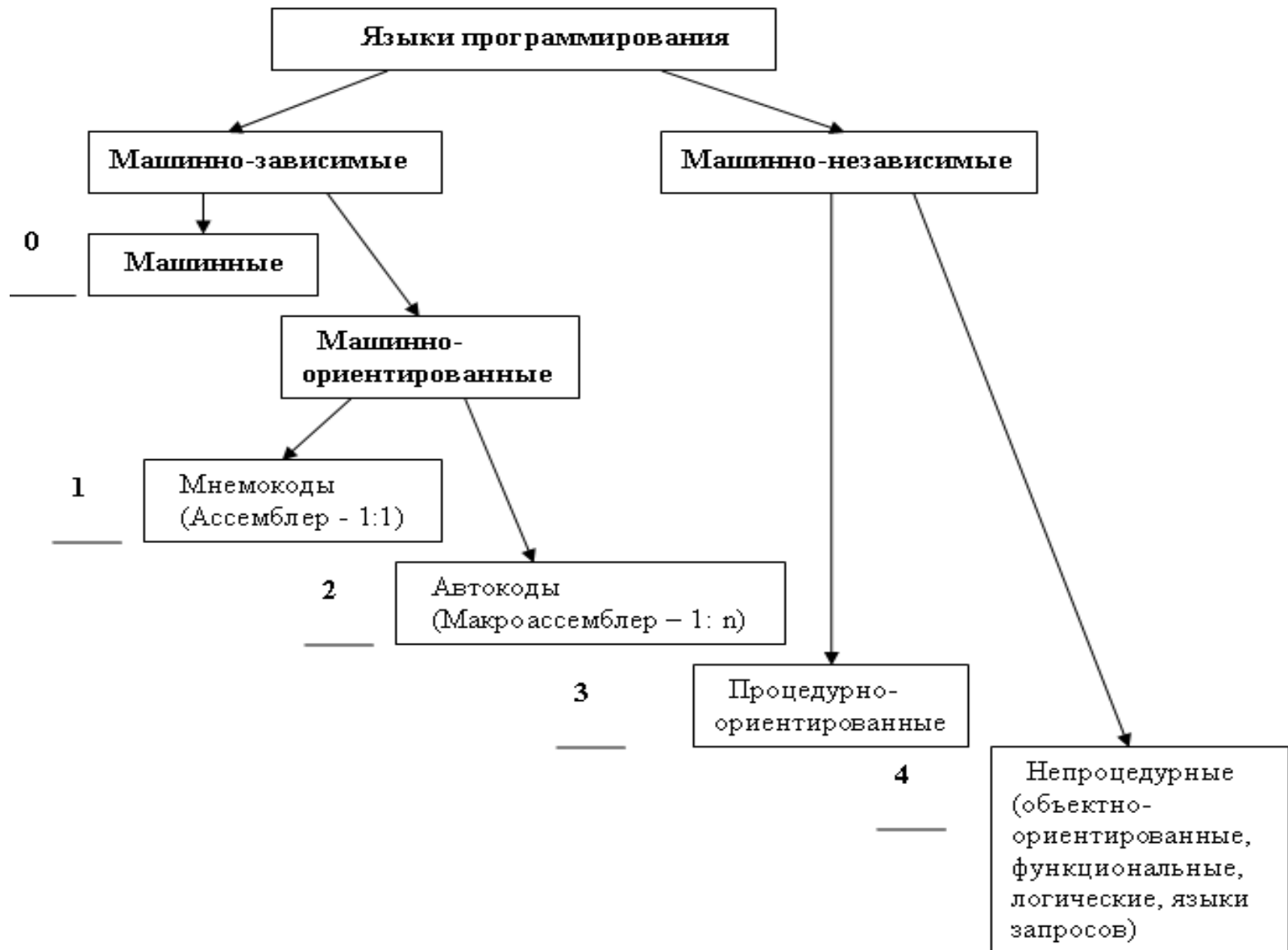
Язык программирования (как и любой другой язык) образуют три его составляющие: **алфавит, синтаксис и семантика**.

Алфавит — это фиксированный для данного языка набор основных символов, т.е. "букв алфавита", из которых должен состоять любой текст на этом языке — никакие другие символы в тексте не допускаются.

Синтаксис — это совокупность правил, которые определяют допустимые последовательности основных символов в языке (правильно построенные фразы).

Семантика — это система истолкования отдельных конструкций языка и целых фраз.

Классификация языков программирования



Классификация языков программирования

Машинные языки и машинно-ориентированные языки – это языки **низкого уровня**, требующие указания мелких деталей процесса обработки данных на уровне машинных операций компьютера.

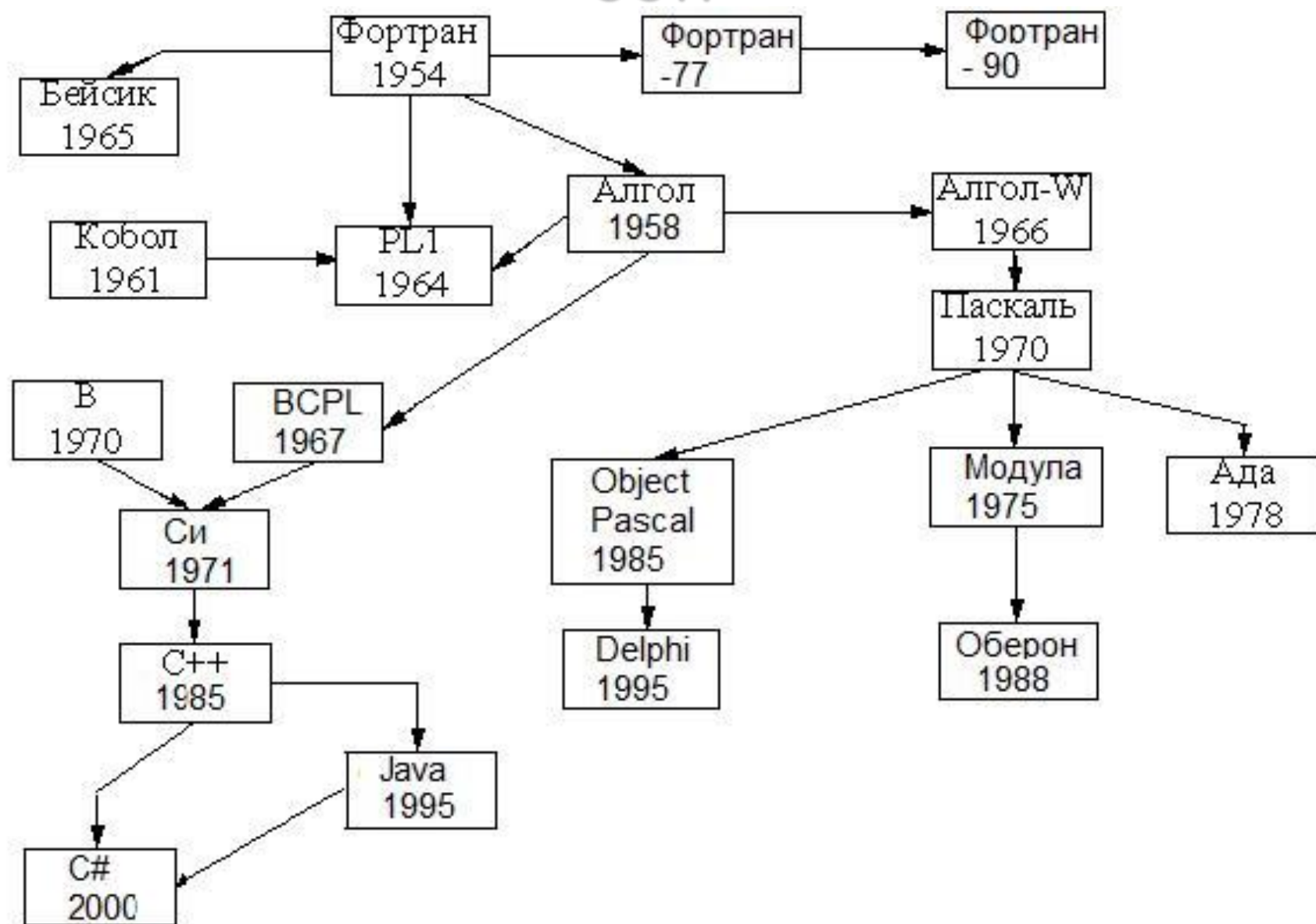
Машинно-независимые языки – это языки **высокого уровня**. Набор средств, допустимых для использования, **не зависит от набора машинных операций**, а выбирается из соображений удобства формулирования процессов решения задач определенного класса.

Примеры языков высокого уровня (3 и 4-е поколения):

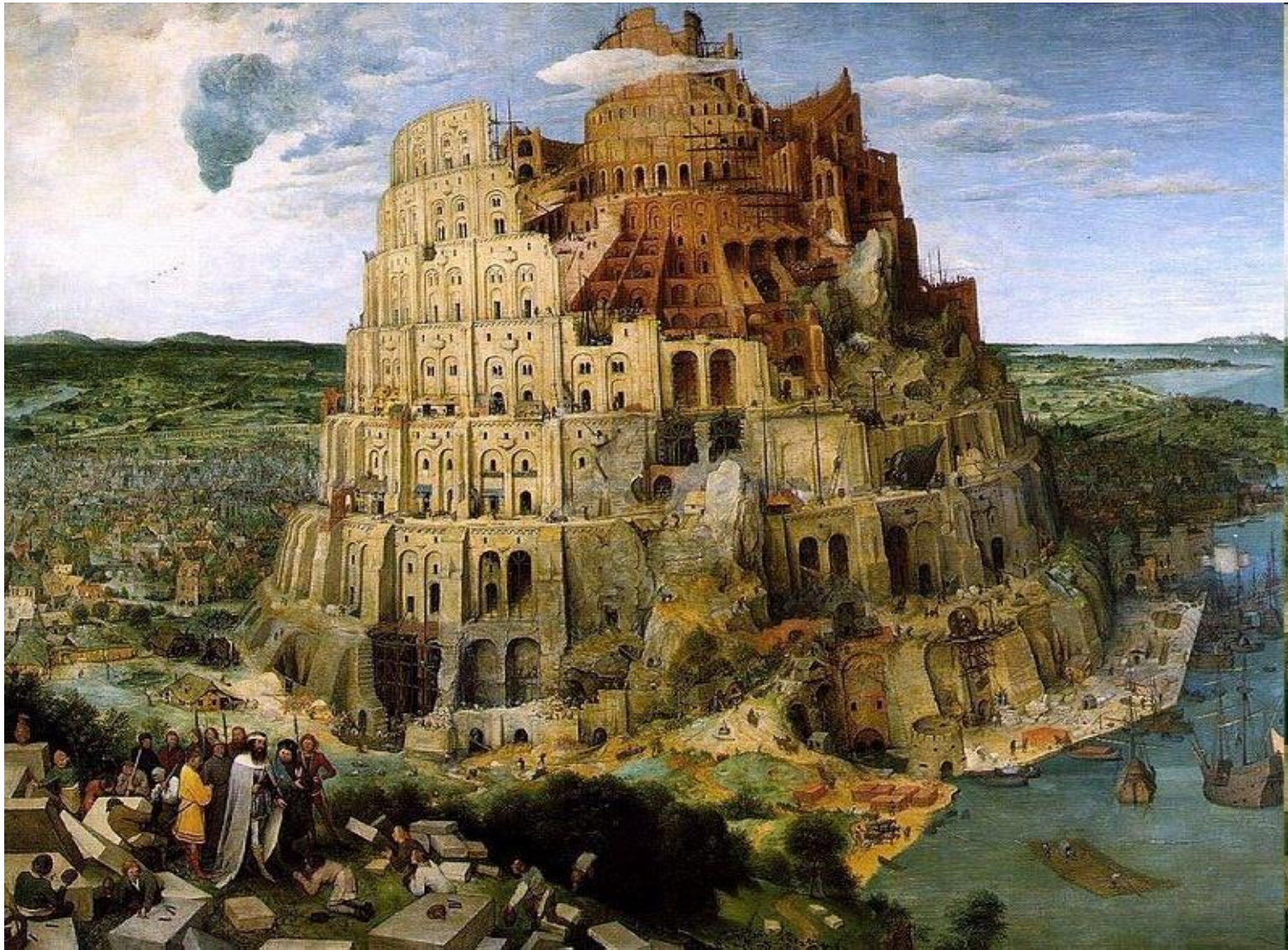
- ✓ **процедурные** (алгоритмические) (Pascal, C, Fortran и др.);
- ✓ **объектно-ориентированные** (Object Pascal, C++, Java, C# и др.);
- ✓ **логические** (Planner, Prolog, QLisp и др.);
- ✓ **функциональные** (Lisp, Scheme и др.).

Языки 5-го поколения (дискуссионное). Создание программ с помощью визуальных средств разработки, без знания программирования. Основная идея была заключена в возможности автоматического формирования результирующего текста на универсальных языках программирования (который необходимо откомпилировать). Инструкции вводятся в компьютер с помощью методов, удобных для человека, не знакомого с программированием

Краткая схема развития процедурных языков и языков ООП

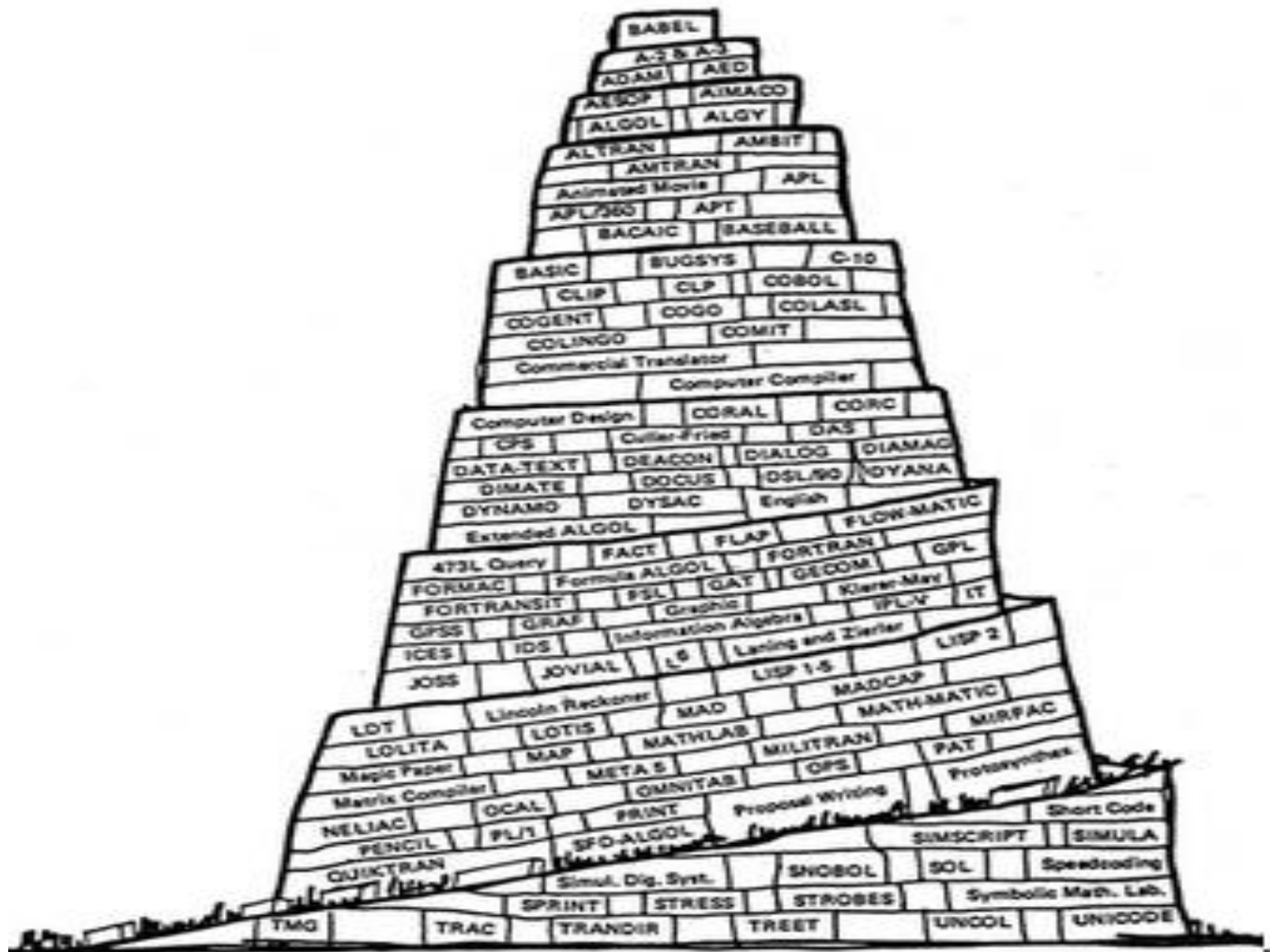


«Вавилонская башня», Питер Брейгель Старший (1563)



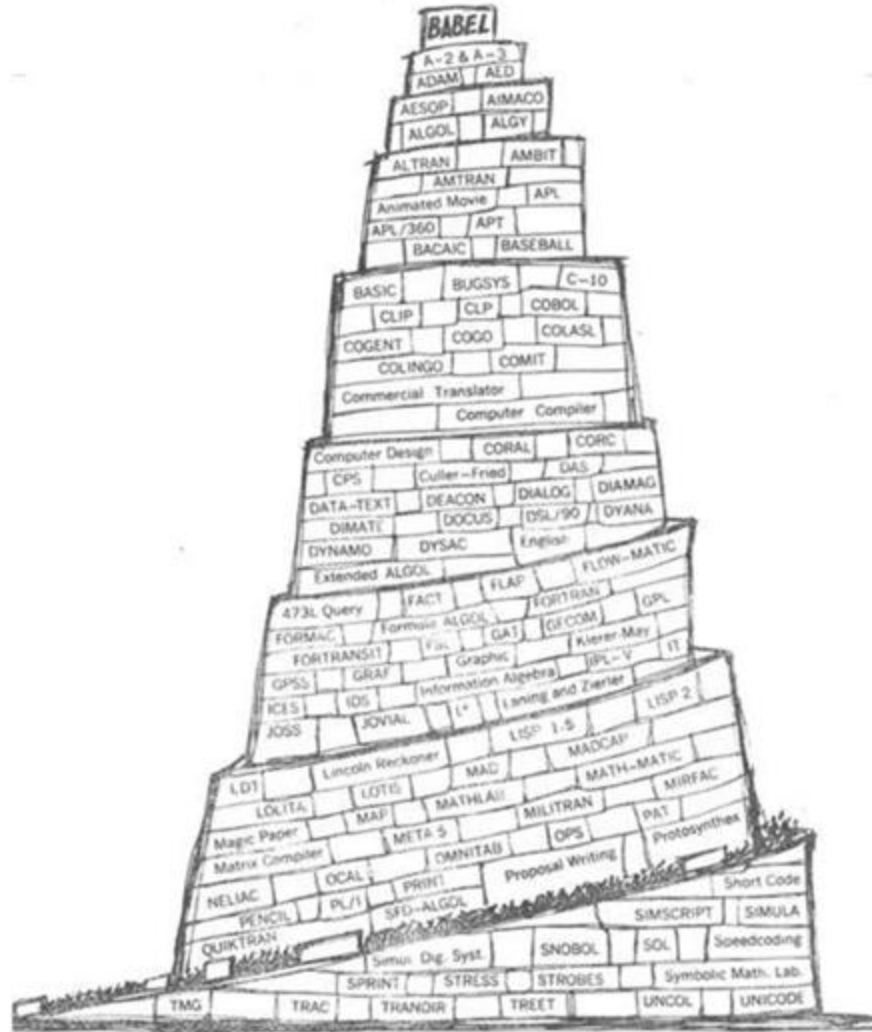
Вавилонская башня языков программирования

<http://generalit.ru/images/it216.png>



Вавилонская башня языков программирования

Языки и системы программирования в 1960-е



Наиболее активный период разработки языков и систем программирования приходится на 1960-е годы.

За это десятилетие в мире родилось более тысячи разнообразных языков, как универсальных, так и специализированных, но выжили и доросли до XXI века дожили немногие, в том числе бессмертные **Fotran**, **Basic**, **Algol**, **Cobol**, **Simula**, **Lisp** и их потомки.

На рисунке: «вавилонская башня» языков программирования, созданных в 1960-е годы

Вавилонская башня языков программирования

Со времени создания первых компьютеров человечество придумало уже более *ВОСЬМИ С ПОЛОВИНОЙ ТЫСЯЧ* языков программирования

Первым общепризнанным языком высокого уровня был **Fortran**, который появилась в ноябре 1954 года. Позднее в октябре 1956 года появилась версия **Fortran I**, а через год **Fortran II**, еще через год вышел **Fortran III**.

Затем появились **Cobol (1957)**, **Lisp (1958)**, **Algol'58 (1958)**, **APL (1960)**

Каждый год появляются новые языки программирования. Некоторыми языками умеет пользоваться только небольшое число их собственных разработчиков, другие становятся известны миллионам людей

Профессиональные программисты иногда применяют в своей работе **более десятка** разнообразных языков программирования