

Языки программирования

Тема 1.



Основные понятия

- Computer
- Software
- Hardware
- Memory
- Processor
- Device
- Program, computer program
- Application software
- System software
- Data
- Information
- Knowledge
- Code, source code
- Programming language
- Translator
- Programming
- Algorithm
- Class
- Encapsulation
- Inheritance
- Polymorphism

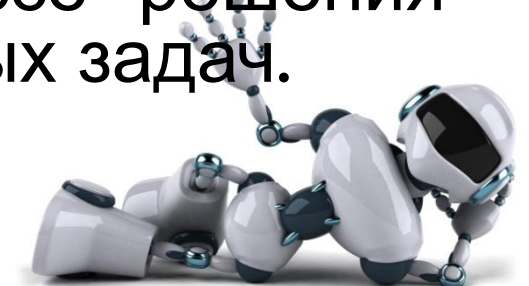
Компьютер. Информация. Программа.



Компьютер

Компьютер - устройство общего назначения, которое может быть запрограммировано, чтобы выполнить конечную, строго определенную последовательность арифметических или логических операций.

Электронная вычислительная машина, ЭВМ - комплекс технических средств, предназначенных для автоматической обработки информации в процессе решения вычислительных и информационных задач.





- Аппаратное
- Программное
- Математические
- Лингвистическое
- Организационное
- Эргономическое
- Информационное

Взаимодействие пользователя с ЭВМ

- Пользователь
- Прикладное ПО
- Системное ПО
- Аппаратное обеспечение

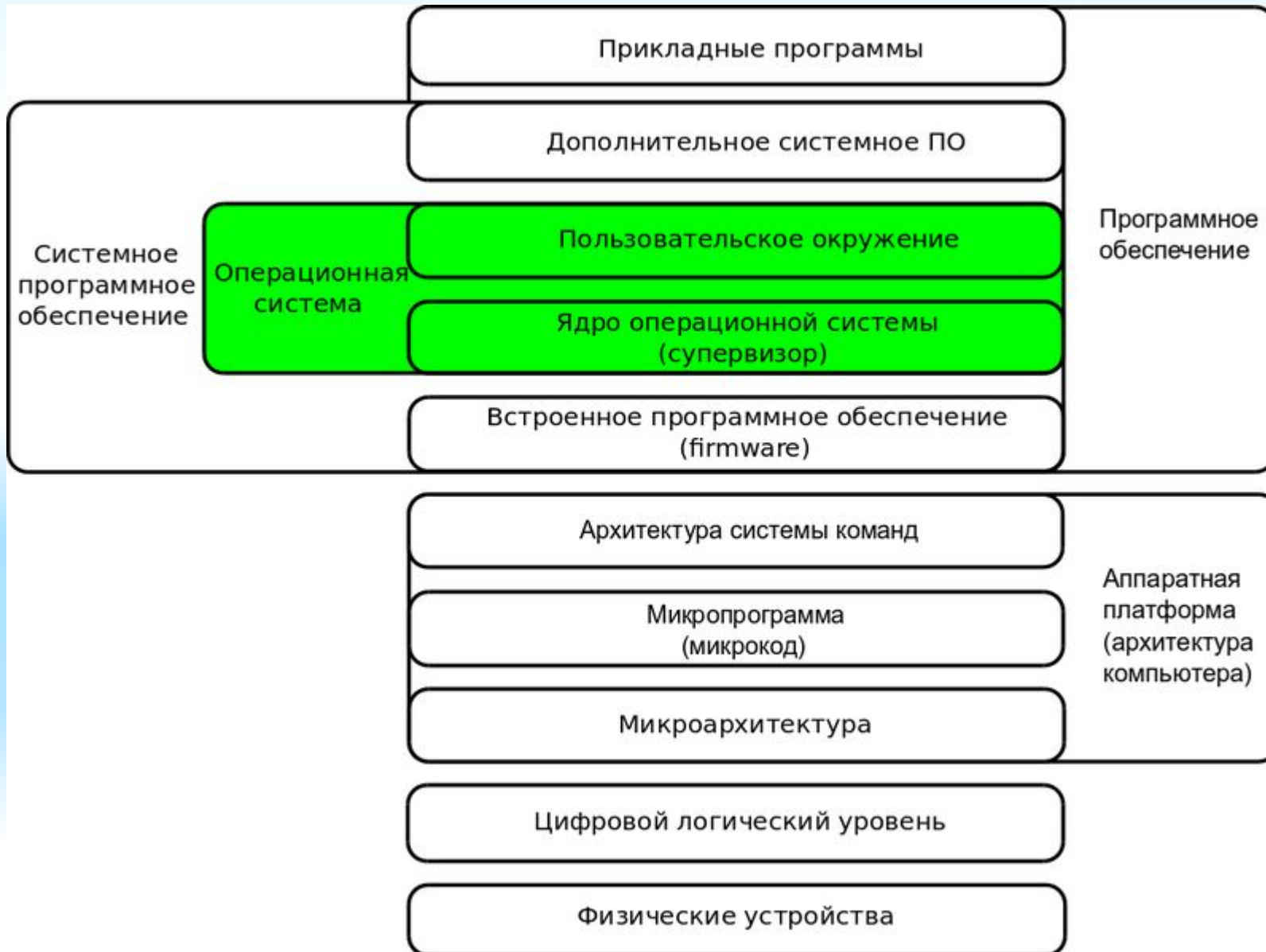




Прикладное ПО

- ПО общего назначения
 - Текстовые редакторы
 - Графические редакторы
 - Табличные редакторы
 - ...
- ПО специального назначения
 - Экспертные системы
 - Мультимедиа приложения
 - Гипертекстовые системы (энциклопедии, справочные системы)
 - ...
- Профессиональное ПО
 - САПР
 - АСУ ТП
 - ГИС
 - Биллинговые системы
 - CRM
 - ERP
 - BI
 - ...

Системное ПО



Состав АО ЭВМ



центральный процессор (CPU);

оперативная память (memory);

устройства хранения информации (storage devices);

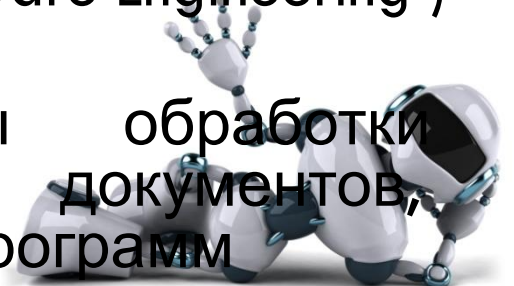
устройства ввода (input devices);

устройства вывода (output devices);

устройства связи (communication devices).

Программное обеспечение

- Все или часть программ, процедур, правил и соответствующей документации системы обработки информации
(ISO/IEC 2382-1: 1993. Information technology).
- Компьютерные программы, процедуры и, возможно, соответствующая документация и данные, относящиеся к функционированию компьютерной системы
(FCD ISO/IEC 24765. Systems and Software Engineering)
- Совокупность программ системы обработки информации и программных документов, необходимых для эксплуатации этих программ
(ГОСТ 19781-90[8]).



Программа

- **Программа** - данные, предназначенные для управления конкретными компонентами системы обработки информации в целях реализации определенного алгоритма.
(ГОСТ 19781.90. ЕСПД. Термины и определения)



Программа

- **Программа** - представленная в объективной форме совокупность данных и команд, предназначенных для функционирования ЭВМ и других компьютерных устройств с целью получения определенного результата, включая подготовительные материалы, полученные в ходе разработки программы для ЭВМ, и порождаемые ею аудиовизуальные отображения.
(ст. 1261 «Программы для ЭВМ» ГК РФ)



- Американский национальный институт стандартов **ANSI** (American National Standards Institute)
- Институт инженеров по электротехнике и электронике **IEEE** (Institute of Electrical and Electronic Engineers)
- Организация международных стандартов **ISO** (International Organization for Standardization)
- Единая система программной документации (**ЕСПД**)

Профессиональные стандарты

1. ANSI
2. IEEE
3. ISO
4. ЕСПД



Подходы к определению информации

- Антропоцентрический
- Техноцентрический
- Закон РФ №24-ФЗ
- Статистическая теория информации
- Кибернетическая теория
- Логико-семантическая теория



Общее в определении информации

- Информация всегда связывается с процессом;
- Информация проясняет ситуацию, т.е. в определенной степени снимает неопределенность (энтропию).



- **Синтаксический** - связан со способом представления информации.
- **Семантический** - передает смысловое содержание информации.
- **Прагматический** - отражает потребительские свойства информации.

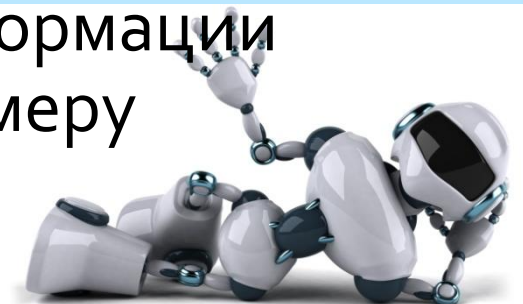
Аспекты рассмотрения информации

1. Синтаксический
2. Семантический
3. Прагматический



Проблемы

- Определение совокупности возможных альтернатив развития событий, процессов или явлений.
- Количественная оценка вероятности этих альтернатив и их значимости для субъекта.
- Введение меры семантической информации (знания, имеющего определенную меру ценности для кого-либо).



Информация

Это совокупность данных о каком-либо объекте, событии, процессе или явлении, передаваемые прямо или косвенно от источника к потребителю, в результате чего снимается или уменьшается неопределенность сложившейся ситуации.



- **Данные** – это представление фактов и идей в формализованном виде, пригодном для передачи и обработки в некотором информационном процессе.
- **Знания** – это закономерности предметной области (принципы, связи, законы), полученные в результате практической деятельности и профессионального опыта, позволяющие специалистам ставить и

Данные и знания

1. Данные
2. Знания



Язык программирования

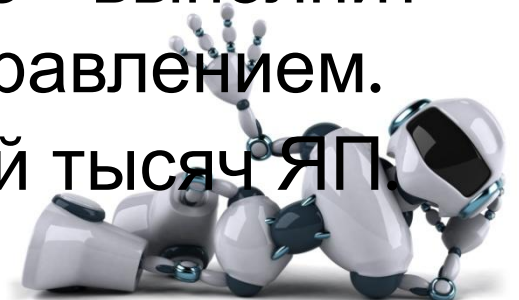


Язык программирования

Язык программирования - формальная знаковая система, предназначенная для записи компьютерных программ.

Язык программирования определяет набор лексических, синтаксических и семантических правил, задающих внешний вид программы и действия, которые выполнит исполнитель (компьютер) под ее управлением.

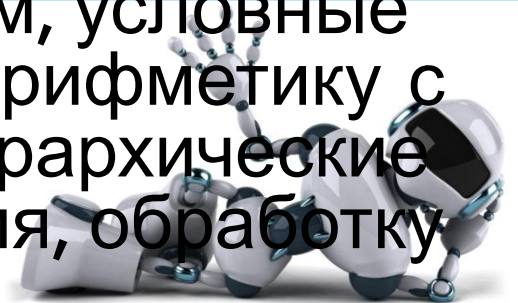
Существует более двух с половиной тысяч ЯП.



История языков программирования

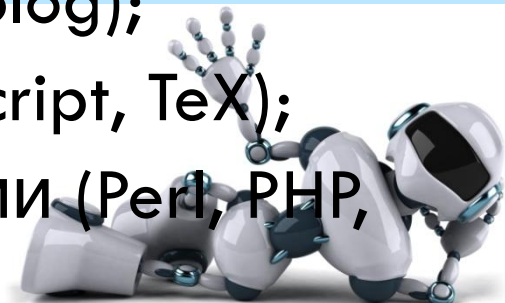
Планкалкюль (нем. *Plankalkül*), - первый в мире высокоуровневый язык программирования, созданный немецким инженером Конрадом Цузе в 1945 году для программирования компьютера Z4.

Планкалкюль поддерживал операции назначения, вызов подпрограмм, условные операторы, итерационные циклы, арифметику с плавающей запятой, массивы, иерархические структуры данных, утверждения, обработку исключений и др.



Области применения ЯП

- научные вычисления (языки C++, FORTRAN, Java);
- системное программирование (языки C++, Java);
- обработка информации (языки C++, COBOL, Java);
- искусственный интеллект (LISP, Prolog);
- издательская деятельность (Postscript, TeX);
- удаленная обработка информации (Perl, PHP, Java, C++);



Факторы развития ЯП

- Предметная ориентация языка ;
- Наличие среды программирования, поддерживающей разработку приложений на конкретном языке программирования;
- Удобство сопровождения и тестирования программ;
- Четкость и логичность синтаксических конструкций языка;
- Стоимость разработки с применением конкретного языка программирования



Машинный код, машинный язык — система команд (инструкций) конкретной вычислительной машины, которая интерпретируется непосредственно процессором или микропрограммами этой ЭВМ.

Каждая инструкция выполняет определённое элементарное действие (# сложение или копирование; в регистре или в памяти) или переход к другому участку кода (изменение порядка исполнения; при этом переход может быть безусловным или условным, зависящим от результатов предыдущих инструкций).

Языки программирования

1. Машинный код
2. ЯП низкого уровня
3. ЯП высокого уровня
4. ЯП сверхвысокого уровня



Язык программирования низкого уровня - язык программирования, близкий к программированию непосредственно в машинных кодах используемого реального или виртуального (например, Java, Microsoft .NET) процессора.

Для обозначения машинных команд обычно применяется мнемоническое обозначение. Это позволяет запоминать команды не в виде последовательности двоичных нулей и единиц, а в виде осмысленных сокращений слов естественного языка.

Языки программирования

1. Машинный код
2. ЯП низкого уровня
3. ЯП высокого уровня
4. ЯП сверхвысокого уровня



Язык ассемблера - язык программирования низкого уровня, мнемонические команды которого (за редким исключением) соответствуют инструкциям процессора вычислительной системы.

Трансляция программы в исполняемый машинный код производится **программой-транслятором**. Позволяет использовать символические **метки** вместо адресов ячеек памяти, а также **директивы** (команды, не переводящиеся в процессорные инструкции, а выполняемые самим ассемблером).

Языки программирования

1. Машинный код
2. ЯП низкого уровня
3. ЯП высокого уровня
4. ЯП сверхвысокого уровня



ЯП высокого уровня - язык программирования, разработанный для быстроты и удобства использования программистом. Основная черта высокоуровневых языков - это абстракция, то есть введение смысловых конструкций, кратко описывающих такие структуры данных и операции над ними, описания которых на ЯП низкого уровня очень длинны и сложны для понимания.

Языки программирования

1. Машинный код
2. ЯП низкого уровня
3. **ЯП высокого уровня**
4. ЯП сверхвысокого уровня



ЯП высокого уровня

В 1954 году был создан первый **язык высокого уровня** - Фортран (англ. FORTRAN - FORmula TRANslator).

Языки высокого уровня имитируют естественные языки, используя некоторые слова разговорного языка и общепринятые математические символы.



Язык программирования сверхвысокого уровня, (VHLL -very high-level programming language) - язык программирования с очень высоким уровнем абстракции. В отличие от языков программирования высокого уровня, где описывается принцип «как нужно сделать», в сверхвысокоуровневых языках программирования описывается лишь принцип «что нужно сделать».

Языки программирования

1. Машинный код
2. ЯП низкого уровня
3. ЯП высокого уровня
4. ЯП сверхвысокого уровня



Парадигмы программирования



Парадигма программирования

- это способ концептуализации, определяющий организацию вычислений и структурирование работы, выполняемой компьютером. Это система идей и понятий, определяющих стиль написания компьютерных программ, а также образ мышления

Парадигмы программирования

1. Структурное
2. Объектно-ориентированное
3. Логическое
4. Функциональное
5. Агентно-ориентированное



Структурное программирование

Структурное программирование предполагает точно обозначенные управляющие структуры, программные блоки, отсутствие инструкций безусловного перехода (GOTO), автономные подпрограммы, поддержку рекурсии и локальных переменных.

Суть такого подхода заключается в возможности разбиения программы на составляющие элементы.

Алгол(1958), Паскаль(1970), Си(1972).



В рамках структурного программирования появляются процедуры и функции. Функции позволяют создавать модульные программы.

Затем начинают использование **структур** (составных типов данных, построенных с использованием других типов), благодаря которым перешли к классам.

Класс – это структура, которая имеет свои переменные и функции, которые работают с

Парадигмы программирования

1. Структурное
2. Объектно-ориентированное
3. Логическое
4. Функциональное
5. Агентно-ориентированное



- **Инкапсуляция** – это свойство системы, позволяющее объединить данные и методы, работающие с ними, в классе и скрыть детали реализации от пользователя.
- **Наследование** – это свойство системы, позволяющее описать новый класс на основе уже существующего с частично или полностью заимствующейся функциональностью.
- **Полиморфизм** – это свойство системы использовать объекты с одинаковым интерфейсом без информации о типе и внутренней структуре объекта.

Симула (1967), **Smalltalk** (70-е),
C++, **Java**, **C#**

Парадигмы программирования

1. Структурное
2. **Объектно-ориентированное**
3. Логическое
4. Функциональное
5. Агентно-ориентированное



Параллельно со структурным, на математической основе создавались функциональные языки - на базе λ -исчислений, и логические языки – на базе логики предикатов.

- **Lisp** - англ. LISt Processing, 1958.
- **Prolog** - англ. PROgramming in LOGic, 1972

Парадигмы программирования

1. Структурное
2. Объектно-ориентированное
3. Логическое
4. Функциональное
5. Агентно-ориентированное



Агентно-ориентированный подход к программированию - парадигма программирования, в которой основными концепциями являются понятия агента и его поведения, которое зависит от среды, в которой он находится.

Взаимодействие между агентами осуществляется не через сообщения, как в ООП, а посредством изменения среды, в которой они находятся.

распределенная объектная

Парадигмы программирования

1. Структурное
2. Объектно-ориентированное
3. Логическое
4. Функциональное
5. **Агентно-ориентированное**



Агент - программная сущность для выполнения поставленных задач. Обладает поведением, а именно: взаимодействует с внешней сложной и динамично-развивающейся средой, могущей модифицироваться или быть модифицированной другими агентами в зависимости от конкретных условий. *Взаимодействие*

подразумевает: 1. восприятие динамики среды; 2. действия, изменяющие среду; 3. рассуждения в целях интерпретации наблюдаемых явлений, решения задач, вывода заключений и

Парадигмы программирования

1. Структурное
2. Объектно-ориентированное
3. Логическое
4. Функциональное
5. Агентно-ориентированное



Примеры программ



Примеры программ

Программа «Hello, world!» для процессора архитектуры x86 (в шестнадцатеричном представлении побайтно):

```
BB 11 01 B9 0D 00 B4 0E 8A 07 43 CD 10 E2 F9 CD  
20 48 65 6C 6C 6F 2C 20 57 6F 72 6C 64 21
```



Примеры программ

```
; Hello World for 6502 Assembler (C64)
ldy #0
beq in
loop:
jsr $ffd2
iny
in:
lda hello,y
bne loop
rts
hello: .tx "Hello World!"
      .by 13,10,0
```



Примеры программ

```
(* Pascal:*)  
program First (Output);  
begin  
  writeln ('Здравствуй, Мир!')  
end.
```

```
(*PASCAL*)  
PROGRAM FIRST (OUTPUT);  
BEGIN  
  WRITELN('Здравствуй, Мир!')  
END.
```



Примеры программ

Алгол 68:

```
begin
```

```
println(`Здравствуй, Мир!')
```

```
end
```

```
comment Русский Алгол 68 comment
```

```
начало
```

```
    печатать(`Здравствуй, Мир!')
```

```
конец
```

```
comment Еще два представления comment
```

```
(println(`Здравствуй, Мир!'))
```

```
(печатать(`Здравствуй, Мир!'))
```



Примеры программ

```
/*Язык С.*/  
#include <stdio.h>  
int main(void)  
{printf("Здравствуй, Мир!");  
  return 0;}
```



Примеры программ

* Hello World in COBOL

IDENTIFICATION DIVISION.

PROGRAM-ID. HELLO.

ENVIRONMENT DIVISION.

DATA DIVISION.

PROCEDURE DIVISION.

MAIN SECTION.

DISPLAY "Hello World!"

STOP RUN.



Примеры программ

```
// Java:  
public class HelloWorld {  
    public static void main(  
        String[] args) {  
        System.out.println(  
            "Здравствуй, Мир!");  
        }  
    }  
}
```



Примеры программ

Лисп (режим исполнения файла программы):
(Print "Здравствуй, Мир!")

Лисп (диалоговый режим):
[1]> (progn (setq x "Здравствуй, Мир!") x)



Примеры программ

Рефал

```
$ENTRY GO{=<Prout 'Здравствуй, Мир!'>;}
```

Prolog

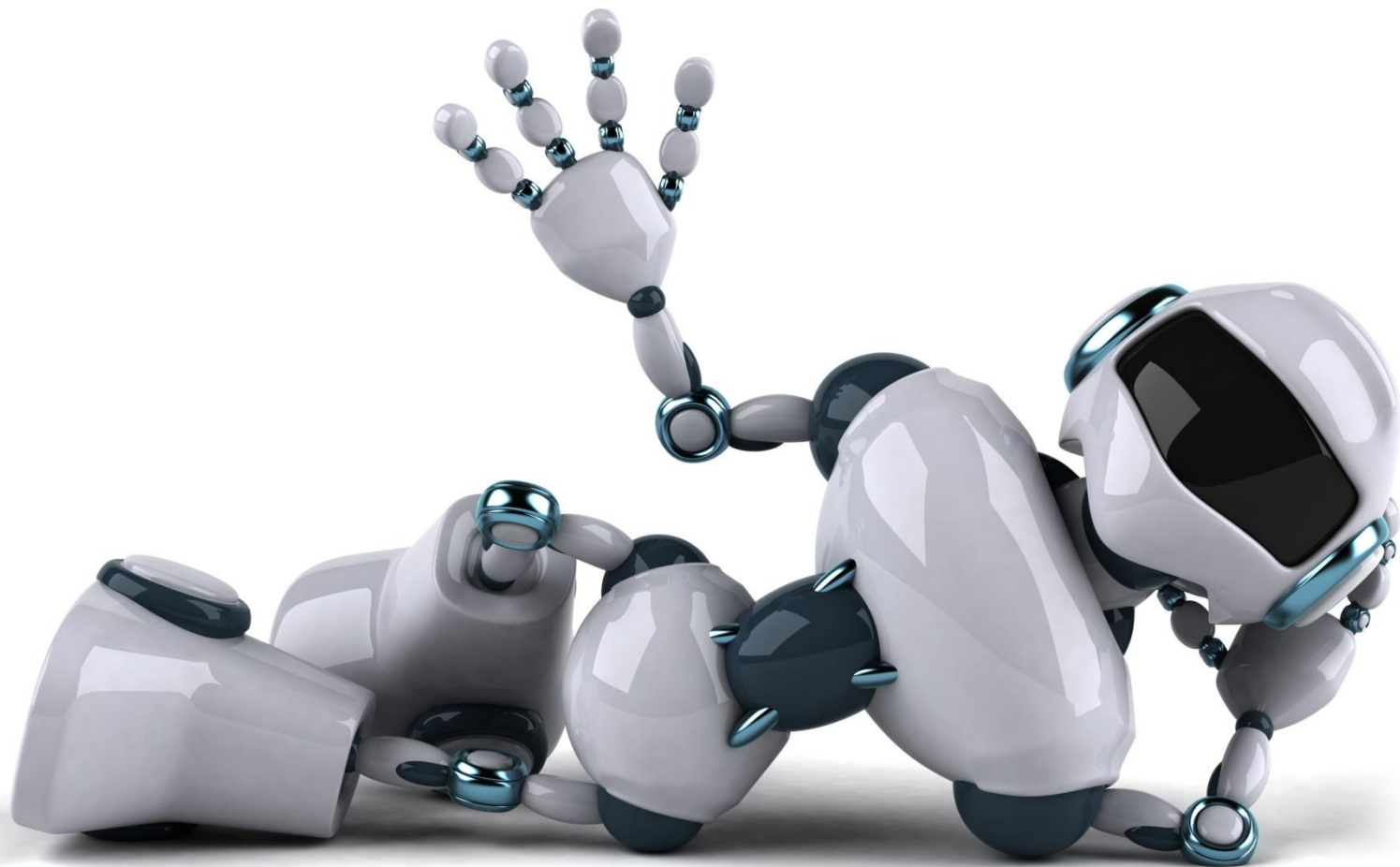
```
:-Print('Здравствуй, Мир!').
```





Основные понятия

- Computer
- Software
- Hardware
- Memory
- Processor
- Device
- Program, computer program
- Application software
- System software
- Data
- Information
- Knowledge
- Code, source code
- Programming language
- Translator
- Programming
- Algorithm
- Class
- Encapsulation
- Inheritance
- Polymorphism



Языки программирования

Тема 1.