



# Машинно-зависимые языки программирования, лекция 8

Каф. ИУ7 МГТУ им. Н. Э. Баумана, 2024 г.



## Java. Java virtual machine (JVM)

Java - объектно-ориентированный язык программирования, разработанный компанией Sun Microsystems.

Приложения Java обычно транслируются в специальный байт-код, поэтому они могут работать на любой компьютерной архитектуре, для которой существует реализация виртуальной Java-машины.

Байт-код Java — набор инструкций, исполняемых виртуальной машиной Java. Каждый код операции байт-кода — один байт. Задействовано ~200 команд.

Является одновременно и стековой, и регистровой.

Java Runtime Environment (JRE) - минимальная реализация виртуальной машины. Включает саму виртуальную машину и библиотеку классов.

javac - компилятор байткода, javar - дизассемблер файлов классов Java



# Java virtual machine - система команд

Группы инструкций:

- загрузка и сохранение (например, ALOAD\_0, ISTORE),
- арифметические и логические операции (например, IADD, FCMPL),
- преобразование типов (например, I2B, D2I),
- создание и преобразование объекта (например, NEW, PUTFIELD),
- управление стеком (например, DUP, POP),
- операторы перехода (например, GOTO, IFEQ),
- вызовы методов и возврат (например, INVOKESTATIC, IRETURN).

# Java virtual machine - структура команд

Возможные суффиксы:

i  
l  
s  
b  
c  
f  
d  
a

```
outer:  
for (int i = 2; i < 1000; i++) {  
    for (int j = 2; j < i; j++) {  
        if (i % j == 0)  
            continue outer;  
    }  
    System.out.println(i);  
}
```

```
0:  iconst_2  
1:  istore_1  
2:  iload_1  
3:  sipush 1000  
6:  if_icmpge 44  
9:  iconst_2  
10: istore_2  
11: iload_2  
12: iload_1  
13: if_icmpge 31  
16: iload_1  
17: iload_2  
18: irem  
19: ifne 25  
22: goto 38  
25: iinc 2, 1  
28: goto 11  
31: getstatic #84; // Field java/lang/System.out:Ljava/io/PrintStream;  
34: iload_1  
35: invokevirtual #85; // Method java/io/PrintStream.println:(I)V  
38: iinc 1, 1  
41: goto 2  
44: return
```



# Платформа .NET. CLR, CIL

.NET (2002) - платформа, основанная на CLR (Common Language Runtime, общезыковая исполняющая среда).

CLR — исполняющая среда для байт-кода CIL (MSIL), в которой компилируются программы, написанные на .NET-совместимых языках программирования.

CIL (Common Intermediate Language) — «высокоуровневый ассемблер» виртуальной машины .NET, основанный на работе со стеком.

```
ldloc.0      // push local variable 0 onto stack
ldloc.1      // push local variable 1 onto stack
add          // pop and add the top two stack items then push the result
              // onto the stack
stloc.0      // pop and store the top stack item to local variable 0
```

ildasm, ilasm - дизассемблер/ассемблер промежуточного языка (intermediate language)

# CIL - пример

```
static void Main ( string [] args )
{
    for ( int i = 2; i < 1000; i++ )
    {
        for ( int j = 2; j < i; j++ )
        {
            if ( i % j == 0 )
                goto outer;
        }
        Console.WriteLine( i );
        outer:;
    }
}
```

```
.assembly primes{}
.method private hidebysig static void Main(string[] args) cil managed
{
    .entrypoint
    .maxstack 2
    .locals init (int32 v_0,
                  int32 v_1)

        ldc.i4.2
        stloc.0 // v_0=2
        br.s     IL_001f
    IL_0004: ldc.i4.2
        stloc.1 // v_1=2
        br.s     IL_0011
    IL_0008: ldloc.0 // v_0
        ldloc.1 // v_1
        rem     // v_0 % v_1
        brfalse.s IL_001b
        ldloc.1 //v_1
        ldc.i4.1
        add
        stloc.1 // v_1+=1
    IL_0011: ldloc.1 //if v_1<=v_0
        ldloc.0
        blt.s     IL_0008
        ldloc.0
        call     void [mscorlib]System.Console::WriteLine(int32)
    IL_001b: ldloc.0 //v_0
        ldc.i4.1
        add
        stloc.0 //v_0+=1
    IL_001f: ldloc.0
        ldc.i4     0x3e8
        blt.s     IL_0004 //if v_0<=1000
        ret
}
```



# WebAssembly (wasm)

WebAssembly — это бинарный формат инструкций для стековой виртуальной машины, предназначенной для компиляции программ на ЯВУ (C, C++, C#, Go, TypeScript/AssemblyScript, Kotlin, Pascal, Rust, D, Erlang) для WEB.



## WebAssembly - пример

Исходный код на C	«линейный ассемблерный байт-код»	бинарный код WASM
<pre>int factorial(int n) {     if (n == 0)         return 1;     else         return n * factorial(n-1); }</pre>	<pre>get_local 0 i64.eqz if i64     i64.const 1 else     get_local 0     get_local 0     i64.const 1     i64.sub     call 0     i64.mul end</pre>	<pre>20 00 50 04 7e 42 01 05 20 00 20 00 42 01 7d 10 00 7e 0b</pre>