

TAHAK pro Javu

Lepe kdyz pouzijete tento overeny - nez kdyz si vyrobite sami spatny.

Prehled operatoru

priorita	operator	typ operandu	asociativita	operace
1	++	aritmeticky	P	pre/post inkrementace
	--	aritmeticky	P	pre/post dekrementace
	-	aritmeticky	P	unarni plus/minus
	~	celociselny	P	bitova inverze
	!	logicky	P	logicka negace
	(typ)	libovolny	P	pretypovani
2	* / %	aritmeticky	L	nasobeni, deleni, zbytek
3	-	aritmeticky	L	scitani, odecitani
	+	String	L	zretezeni
4	<<	celociselny	L	posun vlevo
	>>	celociselny	L	posun vpravo
	>>>	celociselny	L	posun vpravo s doplnovanim nuly
	< <=<, =	aritmeticky	L	porovnani
	instanceof	objekt	L	test tridy
	== !=	primitivni	L	rovno nerovno
7	&	celociselny nebo logicky	L	bitove nebo logicke AND
8	^	-"-	L	bitove nebo logicke XOR
9		-"-	L	bitove nebo logicke OR
10	&&	logicky	L	logicke AND vyhodnocovane zkracene
11		-"-	L	logicke OR vyhodnocovane zkracene
12	? :	-"-	P	podmineny operator
13	= = -= *= /= %= == &= ^= =	libovolny	P	prirazeni

Co je uplne jinak nez v C:

- Java se interpretuje prostrednictvim zkompilovaneho byte code. Interpretaci provadi JVM (Java Virtual Machine) - viz program java.
- Neexistuje linker - vse zarizuje loader.

- Neexistuje delete ani destruktor, tj. dealokaci zarizuje prilezitostne garbage collector.
 - Neni prevod mezi cisly a typem boolean
 - Nejsou unsigned cisla.
-

Zrady:

- Celociselna aritmetika hlasi jen vyjimku pri deleni nulou. Preteceni se nehlasí !!! tedy: 127=-128, 32767=-32768
-

Co jde:

- Deklarace promennych ve tride jsou mozne kdekoli.
 - Dopredne odkazy na metody jsou mozne.
-

Co je trochu jinak:

- Pohybliva (float/double) aritmetika nehlasí nikdy vyjimky - ma zavedeno Infinity a -Infinity a NaN (Not-a-Number).
 - Hvezdicka je jen pro nasobeni a & jen jako logicky soucin.
 - Konstruovat objekt lze jen operatorem new a metodou newInstance tridy Class.
 - Prikazy break a continue mohou odkazovat na navesti.
 - Metoda bez argumentu se definuje xxx() nikoli xxx(void).
 - import se navenek podoba include, ale funguje jinak
 - místo const se uziva final
 - primitivni typy celociselne: **byte**, **short**, **int**, **long**, **char** (1, 2, 4, 8, 2 byte)
 - **pohyblivé: float, double** (4, 8 bytů)
 - **logické: boolean**
-

Co nejde:

- **preprocesor**
 - **struct a union**
 - **typedef**
 - **zjistit velikost objektu, neboť sizeof neexistuje.**
 - **deklarace neexistují - pouze definice**
 - **enum**
 - **goto**
 - **bitova pole**
 - **seznam argumentu metody promenne delky**
 - **pointery**
 - **reference a dereference**
 - **seznam prikazu oddeleny carkou (krome ve for)**
 - **vicenasobne dedeni**
 - **pretizeni operatoru (vyjma znamena : jednak soucet, jednak zretezeni)**
-

Prehled prikazu

vyraz ; - pridaním středníku se z výrazu stane příkaz

{seznam-prikazu}

if (podminka) prikaz

if (podminka) prikaz1 else prikaz2

while (podminka) prikaz

do prikaz **while** (podminka) ; - zakončen středníkem

for (seznam-inicializacnich-vyrazu; podminka ukonceni;

seznam-inkrementacnich-vyrazu) prikaz

switch (vyraz) prikaz

case cons1 : [{...}]

case cons2 : [{...}]

....

[**default**] : [{...}]

};

return [vyraz] ;

synchronized (vyraz) prikaz; vyraz musí být objekt nebo pole

break [navesti] ; ukončení iterace

continue [navesti] ; pokračování v iteraci

try { ... }

catch (Exception e) { ... }

catch (Exception e) { ... }

.....

finally { }

throw objekt typu Exception; vyhození výjimky

import - predefinování možných prefixů metod a proměnných, aby se nemusela vypisovat úplná

jména

package - označení balíku

Modifikatory přístupu

	vlastní balík	vlastní balík	jiny balík	jiny balík
	vlastní třída	jina třída	její podtřída	jina třída
public	V	V	V	V
protected	V	V	V	
	V	V		
private	V			

final - pro class znemožňuje vytvořit podtřidu, u metod znemožňuje predefinovat. Metody private jsou implicitně final. Umožňuje kompilátoru optimalizace .. inlining. Proměnná je konstanta a musí být rovnou inicializována. Je-li inicializátorem konst. výraz, vyhodnotí se již při kompilaci.

native - metoda je implementována v jiném jazyce

synchronized (modifikátor metody) - při provádění metody třídy/objektu se zamkne třída/objekt

volatile - vrtekavá proměnná

static - static proměnná patří třídě, static metoda - metoda třídní metoda. Nedostává implicitně parametr this a proto může používat jen statické proměnné a metody. static inicializátory- při prvním načtení se provedou v pořadí jak jsou uvedeny ve třídě.

abstract - abstract class je ta ktera ma alepon jednu metodu abstraktni. Nelze instanciovat. Abstract metoda ma misto tela strednik. Nemuze byt static, private nebo final. Jeji implemetaci zajisti jen podtrida.

class - definuje tridu:

[modifikatory] class jmenoTridy

[extends jmenoNadtridy] << jen jedna

[implements ... , ...] << i vice

```
{
... promenne a metody ..
}
```

extends - specifikuje nadtridu

implements - indikuje, ze trida ci jedna z jejich natrid implementuje vsechny metody vseh uvedenych interfacu.

interface - deklarace rozhrani. Je vzdy abstraktni:

[public] interface JmenoRozhrani [extends ... , ...] << i vice interfacu ne trid

```
{
```

... nesmi obsahovat: konstruktory, staticke iniciatory tridy,

instancni promenne, metody tridy

... veskere promenne jsou implicitne static final

a musi se iniciovat na konst. vyraz - coz muze byt i objekt vytvoreny pomoci new. Vsechny metody jsou implicitne abstract - tedy bez tela a nemohou byt final, native, static a synchronized.

Umoznuje call-backs.

V public interface jsou vsechny promenne a metody implicitne public.

```
}
```

instanceof - zjistení dedicného vztahu

new - klicove slovo pro konstrukci objektu.

null - prazdna reference

true - hodnota pro typ boolean

false - hodnota pro typ boolean

super - Odkaz na zastinene promenne nadtridy. Umoznuje vyvolani puvodni metody v nadtride.

Typem objektu je typ nadtridy. Je-li jako prvni prikaz v konstrukturu umoznuje vyvolat konstruktor nadtridy. Implicitne super().

this - odkaz na aktualni objekt. Umoznuje lokalni odstineni a predavat ho jako argument statickym metodam nebo metodam jinych trid. Je-li jako prvni prikaz v konstrukturu umoznuje volani jineho konstrukturu teze tridy.

throws - Specifikace vyjimek ktere metoda muze vyhodit a ktere soucasne nejsou podtridami Error ci RuntimeException

[modifikatory] typ jmenoMetody(...,...) throws ... , { ...}

void - vyznacuje ze metoda nic nevraci

finalize - rezervovane jmeno metody. Napise-li se do tridy: void finalize() zavola se ukoncovac tesne pred GC a zavola nakonec i super.finalize().

Co je co:

application - application program, tj. aplikacni program bezne lokalne spousteny prostrednictvim interpretu java.

applet - zdrobnelina application, spoustena browserem nebo appletviewerem zpravidla na dalku s nekterymi omezenimi.

API - Application Programming Interface

AWT - Abstract Windowing Toolkit - zakladni GUI

FTP - File Transfer Protocol

gc - garbage collector

GUI - Graphic User Interface

HTML - HyperText Markup Language

IP - Internet Protocol

IP address - 32 bit id

J2SE - Java 2 Standard Edition

J2EE - Java 2 Enterprise Edition

J2ME - Java 2 Micro Edition

JDK - Java Development Kit

JRE - Java Runtime Environment

JFC - Java Foundation Classes

javac - kompilator Javy

java - interpret Javy

javadoc - dokumentator Javy

jar - kompresor Javy - rozsireny zip

package - balicek trid - zakladni jednotka

SUN - Stanford UNiversity

Swing - vylepsene GUI

TCP - Transport Control Protocol

UDP - Unreliable Datagram Protocol

URL - Uniform Resource Locator