Aula 02

- Instruções de controle

Pauta

- Operadores
- Instruções de controle
- Laços de repetição
- Tipos Primitivos
- Tipos wrapper
- Autoboxing e unboxing

Operadores - atribuição

- Operador de atribuição: =
- Executado da direta para a esquerda

Operadores - atribuição

Operador	Exemplo	Equivalente
+=	c += 7	c = c + 7
-=	d -= 4	d = d - 4
*=	e *= 5	e = e * 5
/=	f /= 3	f = f / 3
%=	g %= 9	g = g % 9

Operadores - Incremento e Decremento

- ++a (pré incremento)
  - Incrementa e só então utiliza seu valor
- a++(pós incremento)
  - Utiliza o valor de a e depois incrementa
- --b
  - Decrementa e só então utiliza seu valor
- b--
  - Primeiro utiliza o valor de b para depois

Operadores comparação

- Igualdade: == (2 sinais de igual)
- Diferença: !=
- Maior, maior igual: >, >=
- Menor, menor igual: <, <=

Operadores lógicos

- Retornam booleano
- || (OU condicional)
- && (E condicional)
- | (OU lógico booleano)
- & (E lógico booleano)
- ^ (OU exclusivo lógico booleano)
- ! (NÃO lógico)

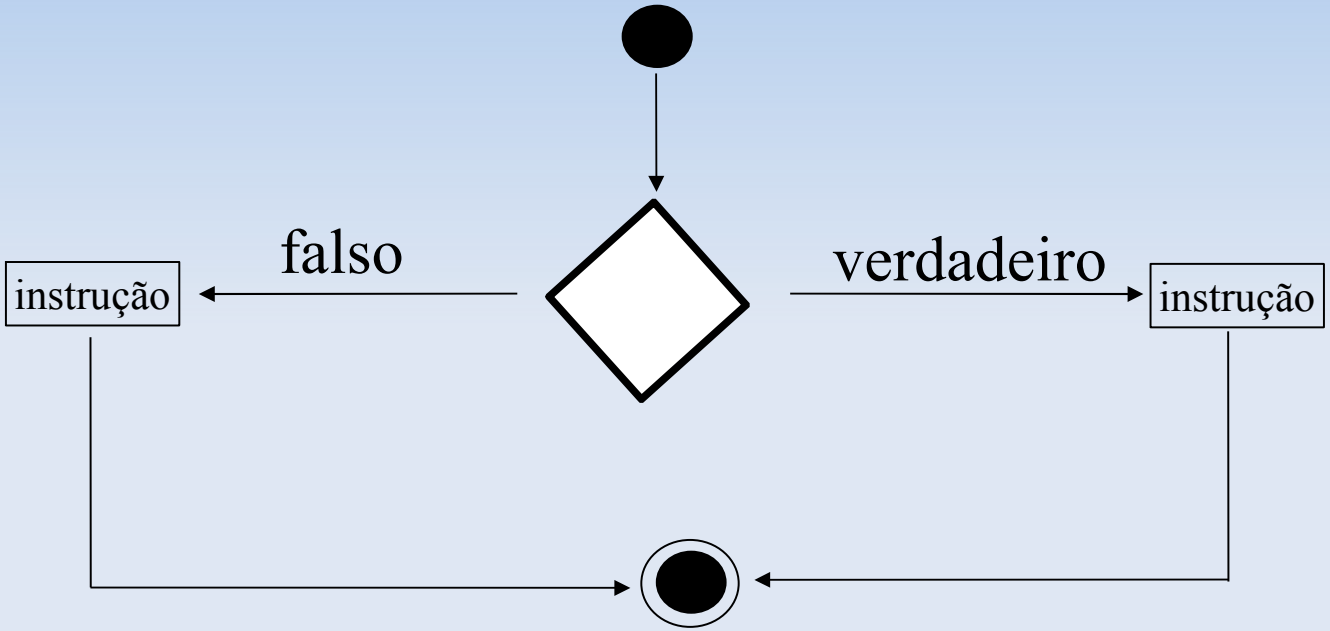
Operadores bit a bit

- << move os bit's para esquerda
- >> move os bit's para direita
- >>> move os bit's para direita completando os espaços com 0

Execução e atribuição

- O Java executa da esquerda para a direita e atribui da direita para a esquerda.

Instruções de controle - if




```
if( condição ){
      ..executa quando verdadeiro
}
else{
    ..executa quanto falso
}
```

Ver código 01 do material de apoio

- ? :

  expressao ? Se verdadeiro : se falso

```
System.out.println( nota > 5 ?“aprovado”:”reprovado”)
```

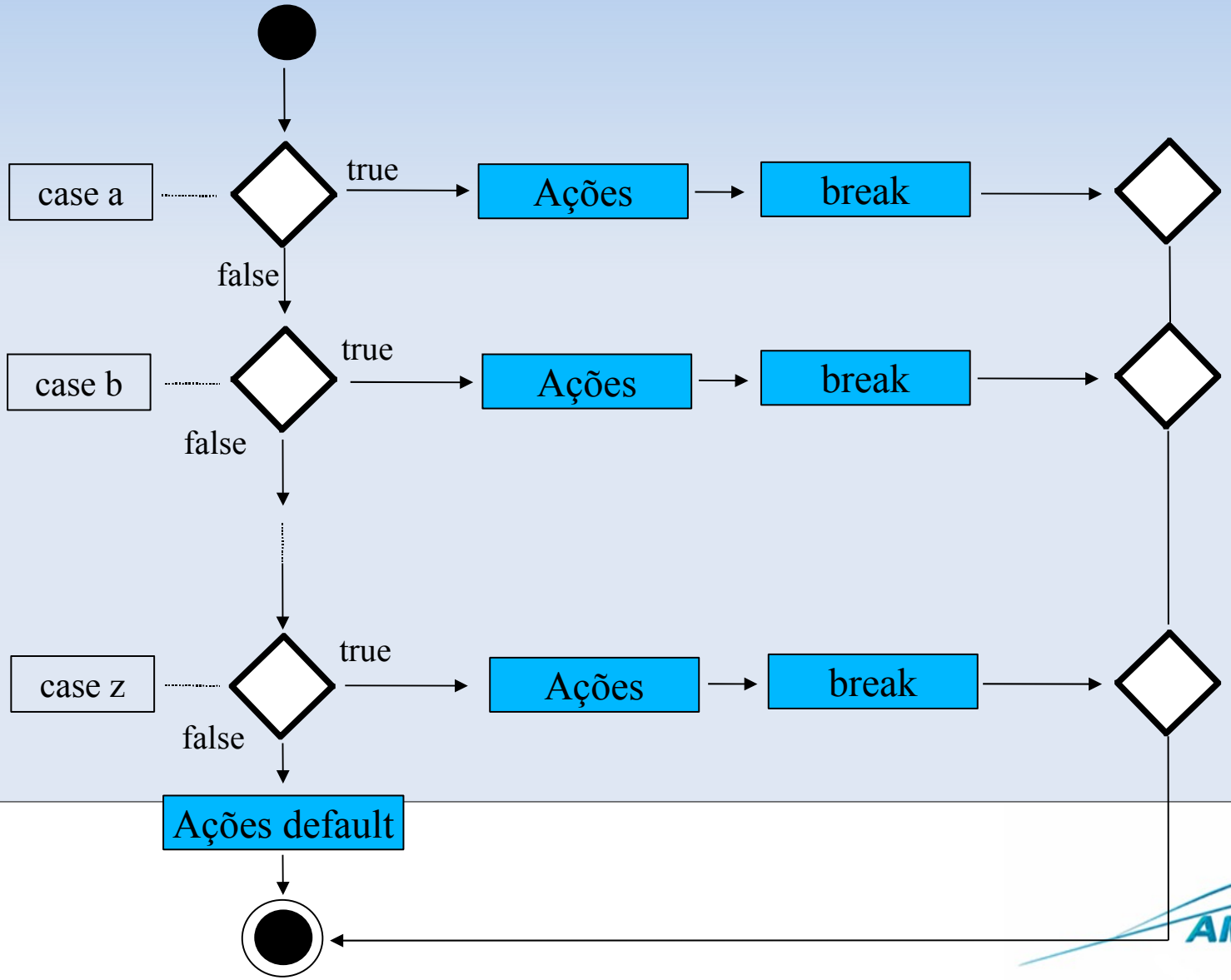
? :

expressao ? Se verdadeiro : se falso

- `System.out.println( nota > 5 ?“aprovado”:”reprovado”)`

Ver código 02 e 03do material de apoio

Estrutura de controle - switch



Estrutura de controle - switch

```
switch( v ){
    case 1:
    ..executa se v for 1
    case 2:
    ..executa se v for 2
    case 3:
    ..executa se v for 3
    default:
    ..executa se v for diferente dos demais
}
```

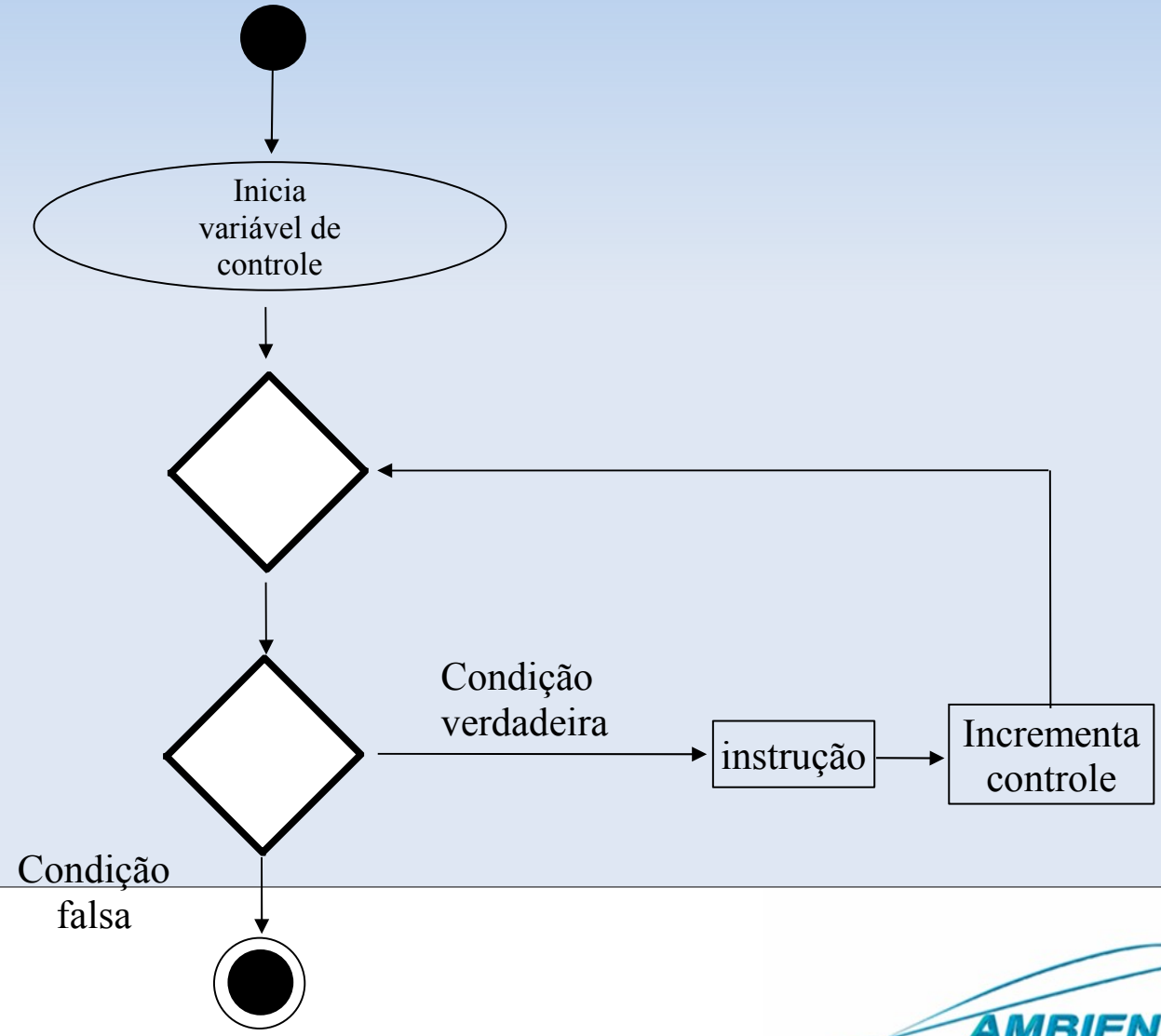
Obs.: o argumento passado para o switch deverá ser int

Estrutura de controle - break

- **break:** quando executado em um *while, for, do...while ou switch* ocasiona a saída imediada desta instrução;

Ver código 04 do material de apoio

Laços de repetição - for

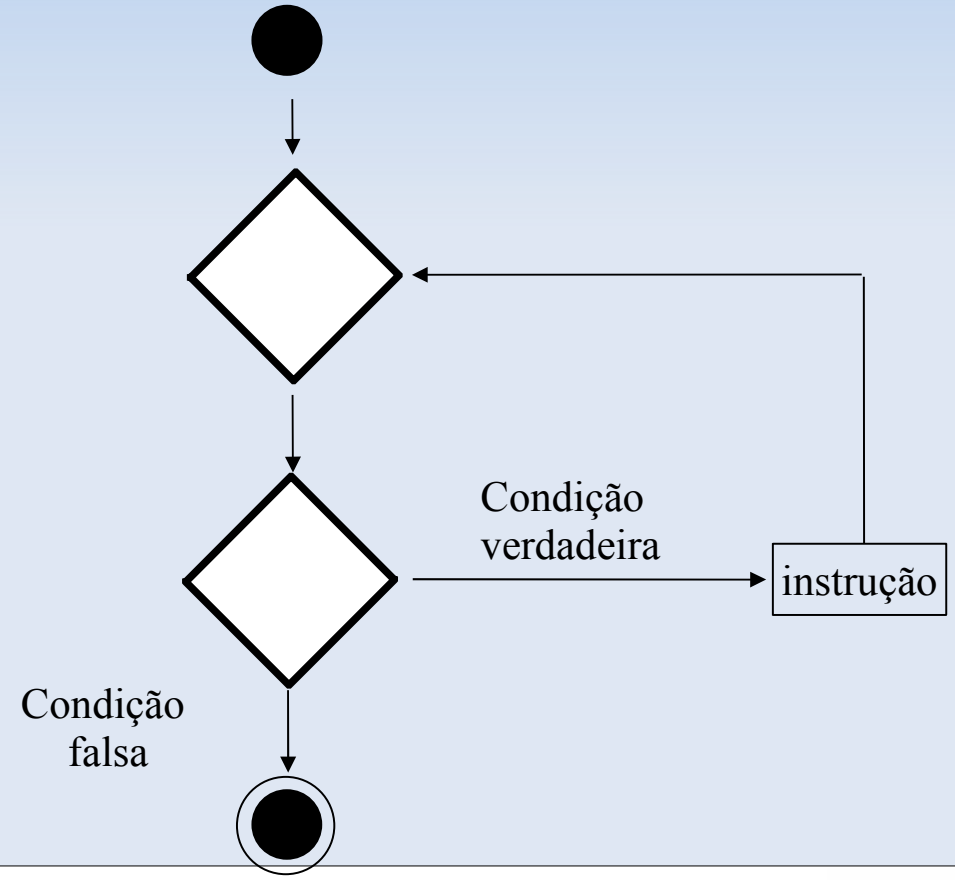


Laços de repetição - for

```
for( inicialização ; condição; incremento ){
    ..executa enquanto condição for
verdadeira
}
```

Ver código 05 do material de apoio

Laços de repetição - while

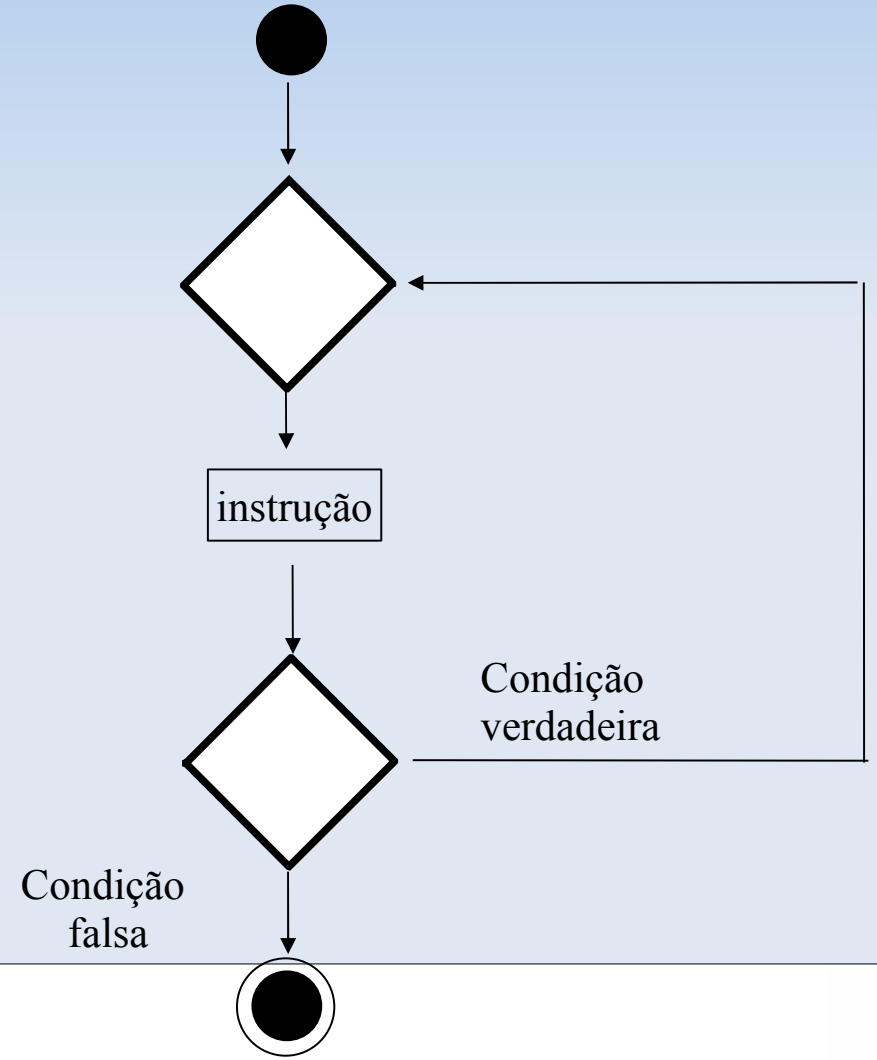


Laços de repetição - while

```
while( condição ){
    ..executa enquanto condição for
verdadeira
}
```

Ver código 06 do material de apoio

Laços de repetição - do ... while



Laços de repetição - do ... while

```
do{
    ..executa enquanto condição for
verdadeira
}while( condição );
```

Ver código 07 do material de apoio

Laços de repetição - continue

- **continue:** quando executado em um *while, for ou do...while* pula as instruções restantes no corpo do loop e prossegue com a próxima iteração do loop.
- Utilizando continue modifique o código da classe ExemploFor para não imprimir a linha 3.

Tipos Primitivos

- Inteiros
- Ponto Flutuante
- Caracter
- booleano

Tipos Primitivos

- Inteiros
  - byte
  - short
  - int
  - long

Tipos Primitivos

Tipo	*Armazenamento*	*Intervalo*
byte	1 byte	-128 a 127
short	2 bytes	-32.768 a 32.767
int	4 bytes	-2.147.483.648 a 2.147.483.647
long	8 bytes	-9.223.372.036.854.775.808L a 9.223.372.036.854.775.807L

Tipos Primitivos

- Ponto Flutuante
  - float
  - double

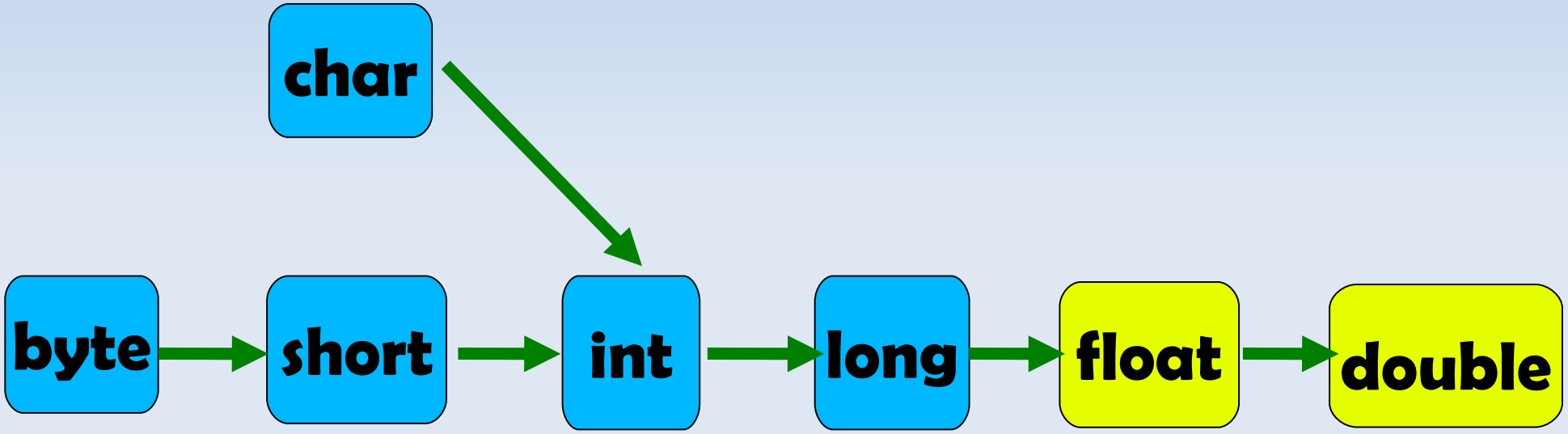
Tipos Primitivos

Tipo	*Armazenamento*	*Intervalo*
float	4 bytes	Aproximadamente +/- 3.40282347E+38F
double	8 bytes	Aproximadamente +/- 1.79769313486231570E+308

Tipos Primitivos

- char
  - \u0000 a \uFFFF ou 0 a 65535
- boolean
  - true ou false

Tipos Primitivos - Conversões automáticas...



Ponto flutuante

inteiro

Tipos wrapper

- Pacote java.lang
- Classes que representam tipos primitivos:
  - Byte (byte)
  - Short (short)
  - Character (char)
  - Integer (int)
  - Long (long)
  - Float (float)
  - Double (double)

Tipos wrapper

- Possui métodos para conversões para e a partir de strings
- Forma de tratar tipos primitivos como objetos
  - Armazenar em coleções
  - Seção de app web
  - Etc

Autoboxing e unboxing

- Disponível a partir do java 5
- Conversão automática de tipos primitivos em tipos wrapper
- Exemplo: Integer numero = 5;

Duvidas??

AMBIENT
INFORMÁTICA

Qual o valor de x e de y após a execução do código:

```
int y = 9, x = 2;
y = ++x - y++/x;
x *= y - 1;
```

Qual o valor de i, j e k após a execução do loop for :

```
for(int i = 0, j = 1, k = 1; j < 5; k = i * j, i++, j++)
```

Qual o valor de x e de y após a execução do código:

```
int y = 9, x = 2;
y = ++x - y++/x;
x *= y - 1;
```

Qual o valor de i, j e k após a execução do loop for :

```
for(int i = 0, j = 1, k = 1; j < 5; k = i * j, i++, j++)
```

Exercícios

Escreva um método que receba dois parâmetros do tipo int, um para a altura e outro para a largura e imprima um retângulo com estes dados:

```
********************
********************
********************
********************
********************
********************
```

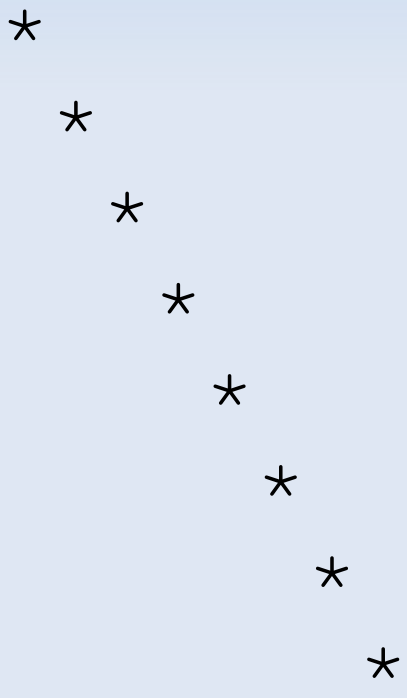
Exercícios

Escreva um método que faça o triângulo abaixo:

```
*
**
***
****
*****
******
*******
```

Exercícios

Escreva um método que faça a reta abaixo:



Fim

AMBIENT
INFORMÁTICA