Programación de Computadores: Java - Enumeraciones, Caracteres y Strings

Juan F. Pérez

Departamento MACC
Matemáticas Aplicadas y Ciencias de la Computación
Universidad del Rosario

juanferna.perez@urosario.edu.co

Segundo Semestre de 2017

Contenidos

Enumeraciones

Enumeraciones

¿Cómo almacenamos información sobre...

y sobre ...

Enumeraciones

¿Cómo almacenamos información sobre...

Nombre

y sobre ...

Enumeraciones

¿Cómo almacenamos información sobre...

- Nombre
- Mayor de edad (sí o no)

y sobre ...

Enumeraciones

¿Cómo almacenamos información sobre...

- Nombre
- Mayor de edad (sí o no)
- Número de hijos

y sobre ...

Enumeraciones

¿Cómo almacenamos información sobre...

- Nombre
- Mayor de edad (sí o no)
- Número de hijos

y sobre ...

- Afiliación a sistema de salud (subsidiado o contributivo)

Enumeraciones

¿Cómo almacenamos información sobre...

- Nombre
- Mayor de edad (sí o no)
- Número de hijos

y sobre ...

- Afiliación a sistema de salud (subsidiado o contributivo)
- Día de la semana (lunes, martes, . . . , domingo)

Enumeraciones

¿Cómo almacenamos información sobre...

- Nombre
- Mayor de edad (sí o no)
- Número de hijos

y sobre ...

- Afiliación a sistema de salud (subsidiado o contributivo)
- Día de la semana (lunes, martes, . . . , domingo)
- Estado civil (soltero, casado, viudo, separado, unión libre)

Enumeraciones

¿Cómo almacenamos información sobre...

- Nombre
- Mayor de edad (sí o no)
- Número de hijos

y sobre ...

- Afiliación a sistema de salud (subsidiado o contributivo)
- Día de la semana (lunes, martes, . . . , domingo)
- Estado civil (soltero, casado, viudo, separado, unión libre)

En este último grupo los datos pertenecen a un conjunto de posibilidades.

Enumeración (cont.)

Enumerar: listar todos los elementos de un conjunto

Enumeración (cont.)

Enumerar: listar todos los elementos de un conjunto

Enumeración: tipo de datos donde los resultados pertenecen a un conjunto que enumeramos completamente

Enumeración (cont.)

Enumerar: listar todos los elementos de un conjunto

Enumeración: tipo de datos donde los resultados pertenecen a un conjunto que enumeramos completamente

Ventajas

Evita trabajar con otras representaciones (String, `int`)

Enumeración (cont.)

Enumerar: listar todos los elementos de un conjunto

Enumeración: tipo de datos donde los resultados pertenecen a un conjunto que enumeramos completamente

Ventajas

Evita trabajar con otras representaciones (String, `int`)

No requiere definiciones adicionales (1 es lunes, 2 es martes, etc., o 1 es subsidiado, 2 es contributivo)

Enumeración (cont.)

Enumerar: listar todos los elementos de un conjunto

Enumeración: tipo de datos donde los resultados pertenecen a un conjunto que enumeramos completamente

Ventajas

Evita trabajar con otras representaciones (String, `int`)

No requiere definiciones adicionales (1 es lunes, 2 es martes, etc., o 1 es subsidiado, 2 es contributivo)

Código más fácil de escribir, leer y extender

Enumeraciones en Java

Revisemos `proyectoEnumeraciones`

Enumeración de días de la semana

```
public enum DiaSemana{
LUNES, MARTES, MIERCOLES, JUEVES, VIERNES, SABADO,
    DOMINGO;
}
```

Enumeraciones en Java

Revisemos `proyectoEnumeraciones`

Enumeración de días de la semana

```
public enum DiaSemana{
LUNES, MARTES, MIERCOLES, JUEVES, VIERNES, SABADO,
    DOMINGO;
}
```

Método para evaluar si el día es fin de semana o no

Enumeraciones en Java (cont.)

```
public void run(){
 DiaSemana hoy = DiaSemana.LUNES;
 DiaSemana ayer = DiaSemana.DOMINGO;
 DiaSemana anteayer = DiaSemana.SABADO;
 DiaSemana manana = DiaSemana.MARTES;

 evaluarFinDeSemana(hoy);
 evaluarFinDeSemana(ayer);
 evaluarFinDeSemana(anteayer);
 evaluarFinDeSemana(manana);
}
```

Enumeraciones en Java (cont.)

```
void evaluarFinDeSemana(DiaSemana dia){
 switch(dia){
 case LUNES:
  println("No es fin de semana");
  break;
 case MARTES:
  println("No es fin de semana");
  break;
 case MIERCOLES:
  println("No es fin de semana");
  break;
 case JUEVES:
  println("No es fin de semana");
break;
 case VIERNES:
  println("No es fin de semana");
  break;
 case SABADO:
  println("Es fin de semana");
  break;
```

Enumeraciones en Java (cont.)

```
void evaluarFinDeSemana2(DiaSemana dia){
 switch(dia){
  case LUNES:
  case MARTES:
  case MIERCOLES:
  case JUEVES:
  case VIERNES:
   println("No es fin de semana");
   break;
  case SABADO:
  case DOMINGO:
   println("Es fin de semana");
   break;
 }
}
```

Enumeraciones en Java (cont.)

```
void evaluarFinDeSemana3(DiaSemana dia){
 switch(dia){
  case SABADO:
  case DOMINGO:
   println("Es fin de semana");
   break;
  default:
   println("No es fin de semana");
 }
}
```

Enumeraciones en Java (cont.)

```
void evaluarFinDeSemana4(DiaSemana dia){
 if (dia == DiaSemana.SABADO || dia == DiaSemana.DOMINGO){
  println("Es fin de semana");
 }
 else{
  println("No es fin de semana");
 }
}
```

Caracteres

Caracteres

Caracteres: tipo de datos primitivo `char`

Caracteres

Caracteres: tipo de datos primitivo `char`

¿Cómo los representamos en el computador?

Caracteres

Caracteres: tipo de datos primitivo `char`

¿Cómo los representamos en el computador?

Sabemos representar enteros (conjuntos de bits, representación binaria)

Caracteres

Caracteres: tipo de datos primitivo `char`

¿Cómo los representamos en el computador?

Sabemos representar enteros (conjuntos de bits, representación binaria)

Usamos enteros: asignamos un entero a cada caracter

Caracteres

Caracteres: tipo de datos primitivo `char`

¿Cómo los representamos en el computador?

Sabemos representar enteros (conjuntos de bits, representación binaria)

Usamos enteros: asignamos un entero a cada caracter

Funciona como una enumeración: no usamos los enteros sino los caracteres directamente (no tenemos que acordamos de los enteros)

Representación de Caracteres

Proyecto `proyectoCaracteres`

Representación de Caracteres

Proyecto `proyectoCaracteres`

Clase `AppletCaracteres.java`

Representación de Caracteres

Proyecto `proyectoCaracteres`

Clase `AppletCaracteres.java`

Ejecutar método `ejemplo1()`

ejemplo1()

```
private void ejemplo1(){
 println("Ejemplo 1");
 char c1 = 'a';
 println("c1: "+c1);
 char c2 = '3';
 println("c2: "+c2);
 char c3 = 3;
 println("c3: "+c3);
 char c4 = 38;
 println("c4: "+c4);
}
```

Representando caracteres

Requerimos asignar a cada caracter un número

Representando caracteres

Requerimos asignar a cada caracter un número

Codificación: asignamos un código a cada caracter

Representando caracteres

Requerimos asignar a cada caracter un número

Codificación: asignamos un código a cada caracter

ASCII: American Standard Code for Information Interchange

Representando caracteres

Requerimos asignar a cada caracter un número

Codificación: asignamos un código a cada caracter

ASCII: American Standard Code for Information Interchange

Usa 1 byte (8 bits) para representar un caracter

Representando caracteres

Requerimos asignar a cada caracter un número

Codificación: asignamos un código a cada caracter

ASCII: American Standard Code for Information Interchange

- Usa 1 byte (8 bits) para representar un caracter
- Hasta 256 caracteres diferentes

Representando caracteres

Requerimos asignar a cada caracter un número

Codificación: asignamos un código a cada caracter

ASCII: American Standard Code for Information Interchange

- Usa 1 byte (8 bits) para representar un caracter
- Hasta 256 caracteres diferentes
- 128 estandarizados

Representando caracteres

Requerimos asignar a cada caracter un número

Codificación: asignamos un código a cada caracter

ASCII: American Standard Code for Information Interchange

- Usa 1 byte (8 bits) para representar un caracter
- Hasta 256 caracteres diferentes
- 128 estandarizados
- Primeros 32 y 128 de control, otros imprimibles

Representando caracteres

Requerimos asignar a cada caracter un número

Codificación: asignamos un código a cada caracter

ASCII: American Standard Code for Information Interchange

- Usa 1 byte (8 bits) para representar un caracter
- Hasta 256 caracteres diferentes
- 128 estandarizados
- Primeros 32 y 128 de control, otros imprimibles
- Check out `https://en.wikipedia.org/wiki/ASCII`

Representando caracteres

Requerimos asignar a cada caracter un número

Codificación: asignamos un código a cada caracter

ASCII: American Standard Code for Information Interchange

- Usa 1 byte (8 bits) para representar un caracter
- Hasta 256 caracteres diferentes
- 128 estandarizados
- Primeros 32 y 128 de control, otros imprimibles
- Check out `https://en.wikipedia.org/wiki/ASCII`
- La siguiente figura tomada de `https://raw.githubusercontent.com/wiki/tomgibara/ascii-table/tables/ascii-table-1.1.png`

Representando caracteres

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0 ^@ NUL NULL	1 ^A SOH START OF HEADING	2 ^B STX START OF TEXT	3 ^C ETX END OF TEXT	4 ^D EOT END OF TRANSM.	5 ^E ENQ ENQUIRY	6 ^F ACK ACKNOWL-EDGE	7 ^G BEL BELL	8 ^H BS BACKSP.	9 ^I HT CHARACT. TAB'TION	10 ^J LF LINE FEED	11 ^K VT LINE TAB'TION	12 ^L FF FORM FEED	13 ^M CR CARRIAGE RETURN	14 ^N SO SHIFT OUT	15 ^O SI SHIFT IN
1	16 ^P DLE DATALINK ESCAPE	17 ^Q DC1 DEVICE CONTROL1	18 ^R DC2 DEVICE CONTROL2	19 ^S DC3 DEVICE CONTROL3	20 ^T DC4 DEVICE CONTROL4	21 ^U NAK NEG.ACK-NOWLEDGE	22 ^V SYN SYNCHR. IDLE	23 ^W ETB END OF TRANS.	24 ^X CAN CANCEL	25 ^Y EM END OF MEDIUM	26 ^Z SUB SUBS-TITUTE	27 ^[ESC ESCAPE	28 ^\ FS INFO. SEP. 4	29 ^] GS INFO. SEP. 3	30 ^^ RS INFO. SEP. 2	31 ^_ US INFO. SEP. 1
2	SPACE	! excl ! EXCLAM. MARK	" quot " QUOT. MARK	# num # NUMBER SIGN	$ dollar $ DOLLAR SIGN	% percnt % PERCENT SIGN	& amp & AMPER-SAND	' apos ' APOS-TROPHE	(lpar (LEFT PAREN.	) rpar) RIGHT PAREN.	* ast * ASTERISK	+ plus + PLUS SIGN	, comma , COMMA	- - HYPHEN-MINUS	. period . FULL STOP	/ sol / SOLIDUS
3	0 0 DIGIT ZERO	1 1 DIGIT ONE	2 2 DIGIT TWO	3 3 DIGIT THREE	4 4 DIGIT FOUR	5 5 DIGIT FIVE	6 6 DIGIT SIX	7 7 DIGIT SEVEN	8 8 DIGIT EIGHT	9 9 DIGIT NINE	: colon : COLON	; semi ; SEMI-COLON	< lt < LS.-THAN SIGN	= equals = EQUALS SIGN	> gt > GR.-THAN SIGN	? quest ? QUEST-ION MARK
4	@ commat @ COMM'IAL AT	A A	B B	C C	D D	E E	F F	G G	H H	I I	J J	K K	L L	M M	N N	O O
5	P P	Q Q	R R	S S	T T	U U	V V	W W	X X	Y Y	Z Z	[lsqb [LEFT SQ. BRACKET	\ bsol \ REVERSE SOLIDUS	] rsqb] RT. SQR. BRACKET	^ hat ^ CIRCUM'X ACCENT	_ lowbar _ LOW LINE
6	` grave ` GRAVE ACCENT	a a	b b	c c	d d	e e	f f	g g	h h	i i	j j	k k	l l	m m	n n	o o
7	p p	q q	r r	s s	t t	u u	v v	w w	x x	y y	z z	{ { L. CURLY BRACKET		| VERTICAL LINE	} } R. CURLY BRACKET	~ ~ TILDE	127 ^? DEL DELETE

ASCII code table including entity references, control codes and Unicode names (1.1) © Tom Gibara July 2014

El código ASCII y Unicode

ASCII: diseñado para el idioma inglés

El código ASCII y Unicode

ASCII: diseñado para el idioma inglés

Unicode: mayor número de caracteres

El código ASCII y Unicode

ASCII: diseñado para el idioma inglés

Unicode: mayor número de caracteres

El código ASCII y Unicode

ASCII: diseñado para el idioma inglés

Unicode: mayor número de caracteres

Usa 2 bytes (16 bits) para representar un caracter

El código ASCII y Unicode

ASCII: diseñado para el idioma inglés

Unicode: mayor número de caracteres

- Usa 2 bytes (16 bits) para representar un caracter
- Hasta 65535 caracteres diferentes

El código ASCII y Unicode

ASCII: diseñado para el idioma inglés

Unicode: mayor número de caracteres

- Usa 2 bytes (16 bits) para representar un caracter
- Hasta 65535 caracteres diferentes
- Los primeros 128 son iguales a los ASCII

El código ASCII y Unicode

ASCII: diseñado para el idioma inglés

Unicode: mayor número de caracteres

- Usa 2 bytes (16 bits) para representar un caracter
- Hasta 65535 caracteres diferentes
- Los primeros 128 son iguales a los ASCII
- Check out `https://en.wikipedia.org/wiki/List_of_Unicode_characters`

Caracteres especiales

\b: backspace

\n: nueva línea

\r: retorna al inicio de la línea actual

\t: tabulador

\\: el caracter \

\': el caracter comillas sencillas

\": el caracter comillas dobles

ejemplo2()

```
private void ejemplo2(){
 println("Ejemplo 2");
 char c;
 for (int i = 0; i < 96; i++){
  c = (char)(32 + i);
  println("char representado por entero "+(32 + i)+": "+c);
 }
}
```

Más sobre la Representación de Caracteres

Codificación: enumeración

Más sobre la Representación de Caracteres

Codificación: enumeración

Es posible comparar caracteres

Más sobre la Representación de Caracteres

Codificación: enumeración

Es posible comparar caracteres

Es posible realizar otras operaciones con caracteres

Más sobre la Representación de Caracteres

Codificación: enumeración

Es posible comparar caracteres

Es posible realizar otras operaciones con caracteres

- Algunas (usualmente sumas y restas) pueden tener sentido

Más sobre la Representación de Caracteres

Codificación: enumeración

Es posible comparar caracteres

Es posible realizar otras operaciones con caracteres

- Algunas (usualmente sumas y restas) pueden tener sentido
- Otras no

Más sobre la Representación de Caracteres

Codificación: enumeración

Es posible comparar caracteres

Es posible realizar otras operaciones con caracteres

- Algunas (usualmente sumas y restas) pueden tener sentido
- Otras no
- Manejo cuidadoso

ejemplo3()

```
private void ejemplo3(){
 println("Ejemplo 3");
 char c1 = 'a';
 char c2 = 'b';
 println( c1 + " es " + (c1<c2?"menor":"mayor") + " que " +
    c2  );
 c2 = 'A';
 println( c1 + " es " + (c1<c2?"menor":"mayor") + " que " +
    c2  );
 c2 = '#';
 println( c1 + " es " + (c1<c2?"menor":"mayor") + " que " +
    c2  );
 c2 = '}';
 println( c1 + " es " + (c1<c2?"menor":"mayor") + " que " +
    c2  );
 c2 = (char)(c1 - 32);
 println( c1 + " - 32 es " + c2  );
}
```

Strings

Strings

Cadenas de caracteres

Strings

Cadenas de caracteres

No es un tipo primitivo

Strings

Cadenas de caracteres

No es un tipo primitivo

Es una clase de objetos definida en el paquete `java.lang`

Strings

Cadenas de caracteres

No es un tipo primitivo

Es una clase de objetos definida en el paquete `java.lang`

Constructores:

Strings

Cadenas de caracteres

No es un tipo primitivo

Es una clase de objetos definida en el paquete `java.lang`

Constructores:

```
String cadena1 = "MACC";
```

Strings

Cadenas de caracteres

No es un tipo primitivo

Es una clase de objetos definida en el paquete `java.lang`

Constructores:

```
String cadena1 = "MACC";
String cadena2 = new String("MACC");
```

Strings

Cadenas de caracteres

No es un tipo primitivo

Es una clase de objetos definida en el paquete `java.lang`

Constructores:

```
String cadena1 = "MACC";
String cadena2 = new String("MACC");
```

Taller 4