Introduction to Computers and Programming

Prof. I. K. Lundqvist

Lecture 2
Mar 12 2004

Characters and Strings

- Related Packages
- Operations on both types and the differences between them
- Manipulation of Strings

Concept Question: What is the Output?

1. **1 , 1**
2. **1, 49**
3. **Don’t know**

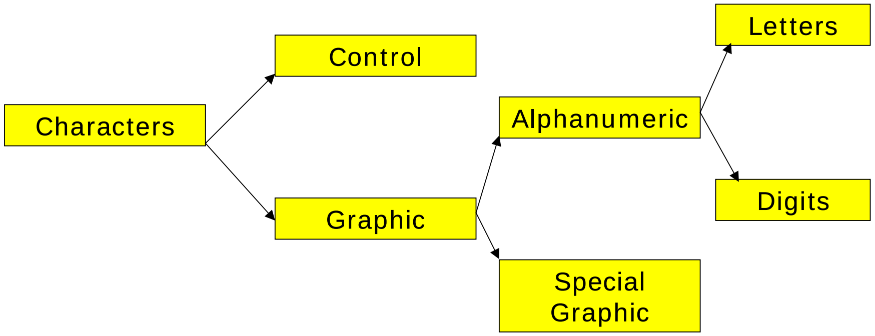
Self_help_chars.adb

Character and Wide Character

- **Character** : correspond to the 256 code positions of Row 00 (also known as **Latin-1**) of the *ISO 10646 Basic Multilingual Plane* (*BMP*).
- **Wide_Character** : correspond to the 65536 code positions of the ISO 10646 Basic Multilingual Plane (BMP).
- ***Note***: First 256 values of Wide_Character have the same name as defined for Character

Ada.Characters.Handling

- Character Classification
- Conversion Functions (both character and string)



Character Handling

- **function** To_Lower (Item : **in** Character) **return** Character;
- **function** To_Upper (Item : **in** Character) **return** Character;
- **function** Is_Character (Item : **in** Wide_Character) **return** Boolean;

```
subtype ISO_646 is Character range
Character'Val(0) ..
Character'Val(127);
```

My_character_converter.adb

ASCII

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2	SP	!	“	#	$	%	&	‘	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

Character Handling

- The predefined operators for the type Character are the same as for any enumeration type
- **function** Is_ISO_646 (Item : **in** Character) **return** Boolean;
- **function** To_ISO_646 (Item : **in** Character; Substitute : **in** ISO_646 := ' ') **return** ISO_646;

Demo_char_wide_char.adb

Strings

- A string is an array of characters (static)
- So, a *string* S consists of the characters: *S*[1], …,*S*[*n*-1], S[*n*]

S_1	S_2		…	S_{n-1}	S_n

- A contiguous subset of the characters of *S* is called a **substring** of *S*
  - I.e., if $1 \leq i \leq j \leq n$ then *S*[*i*], *S*[*i*+1], …, *S*[*j*] is a substring of *S*
- The **null string** contains no characters (“”)

Ada.Strings

- **subtype** Positive **is** Integer **range** 1 .. Integer'Last;
- **type** String **is array** (Positive **range** <>) **of** Character;
- **type** Wide_String **is array** (Positive **range** <>) **of** Wide_Character;

Demo_string_creation.adb

Basic Operations On Strings

- **append**: adds a character to the end of a string
- **insert**: inserts a string in the middle of another string
- **delete**: deletes part of a string
- **concatenate**: joins two strings together
- **substring**: returns part of a string
- **find**: returns the position at which one string occurs within another, or whether it exists
- **length**: returns the number of characters in a string
- **equals**: tests two strings for equality

Pre-defined Operations

- Strings have the same operators as one-dimensional arrays i.e.
  - Concatenation operator &
  - Ordering operators <, <=, >, and >=

```
Question : constant String := "How many characters?";
 -- Question'First = 1, Question'Last = 20
 -- Question'Length = 20 (the number of characters)
Ask_Twice : String := Question & Question;
 -- constrained to (1..40)
```

Demo_Pre_defined_Operations.adb

Ada.Strings.Fixed

- **procedure** Move (Source : **in** String; Target : **out** String; Drop : **in** Truncation := Error; Justify : **in** Alignment := Left; Pad : **in** Character := Space);

- **function** Insert (Source : **in** String; Before : **in** Positive; New_Item : **in** String) **return** String;

- **function** Delete (Source : **in** String; From : in Positive; Through : **in** Natural) **return** String;

Demo_string_fixed.adb