CSCI1530 Computer Principles and Java Programming

Tutorial 8
String Class

Content

- Useful Methods of String Class
  - length/charAt/indexOf/startsWith/ substring
- Examples:
  - SID extraction/reverse
  - Palindrome
  - HKID check digit

String class

- Enclosed by a pair of double-quote character (")
  - e.g. : "12345", "Hello World", "\n"

- Java String class provides many methods

- Document

  http://docs.oracle.com/javase/7/docs/api/java/lang/String.html

int length()

- Return the length of this string
- Example

```
String str = "tutorial";

System.out.println("Length of "+str+": "+str.length());
```

```
Length of tutorial: 8
```

char charAt(int index)

- Return the char value at the specified index
  - Index range: [0, length()-1]
- Example

```java
String str = "tutorial";

System.out.println("Char at 3: "+str.charAt(3));
```

```
Char at 3: o
```

- Index table

Index	0	1	2	3	4	5	6	7
str	t	u	t	o	r	i	a	l

int indexOf(String str)

- Returns the index within this string of first occurrence of the specified substring

```
String str = "tutorial";

System.out.println("Index of 'tor': "+str.indexOf("tor"));
System.out.println("Index of 'tar': "+str.indexOf("tar"));
```

```
Index of 'tor': 2
Index of 'tar': -1
```

- Index table

Index	0	1	2	3	4	5	6	7
str	t	u	t	o	r	i	a	l

`boolean startsWith(String prefix)`

- Test if this string starts with the specified prefix

```
String str = "tutorial";

System.out.println(str.startsWith("tutor"));
```

```
true
```

- Using `indexOf()` to implement a `startsWith()`
  - `return str.indexOf(prefix) == 0`

- In the opposite, to check the ending suffix
  - `boolean endsWith(String suffix)`

String substring(int beginIndex)

- Return a new string that is a substring of this string
  - Substring begins with the character at the specified index and extends to the end of this string.
  - Index starts from 0

```
String str = "tutorial";
System.out.println(str.substring(2));
```

```
torial
```

- Index table

Index	0	1	2	3	4	5	6	7
str	t	u	t	o	r	i	a	l

String substring(int beginIndex, int endIndex)

- Return a new string that is a substring of this string
  - Substring: [beginIndex, endIndex-1]
  - Index starts from 0

```
String str = "tutorial";

System.out.println(str.substring(0, 5));
```

```
tutor
```

- Index table

Index	0	1	2	3	4	5	6	7
str	t	u	t	o	r	i	a	l

More methods in String

Methods	Usage	Returns
toUpperCase()	`"Tutorial".toUpperCase()`	"TUTORIAL"
toLowerCase()	`"Tutorial".toLowerCase()`	"tutorial"
trim()	`"  Tutorial  ".trim()`	"Tutorial"
replace()	`"Tutorial".replace("Tutor", "Potent")`	"Potential"

Example: SID extraction

- to determine if the **sid** conforms to one of these patterns: [11550*****]

```
Scanner scanner = new Scanner(System.in);
String sid = scanner.nextLine();
int flag = sid.indexOf("11550");
if(flag != -1){
    System.out.println(sid+" satisfy the pattern.");
}
else
    System.out.println(sid+" don’t satisfy the pattern.");
```

No need to do any calculation

```
1155010000
1155010000 satisfy the pattern.
1144756921
1144756921 don’t satisfy the pattern.
```

Example: SID reverse

- print the **sid** in reverse

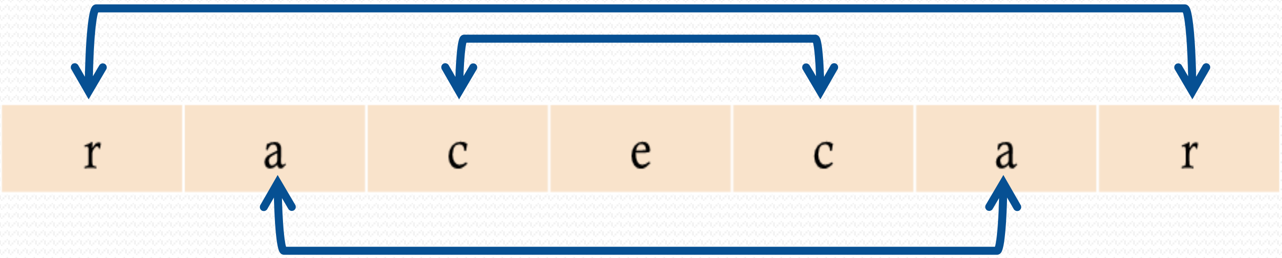
```
Scanner scanner = new Scanner(System.in);
String sid = scanner.nextLine();
int length = sid.length();  //get the length of sid
System.out.print("the reverse of "+sid+": ");
for(int i = length-1; i>=0; i--){
    System.out.print(sid.charAt(i));
}
```

Index belongs to
[0,length-1]

```
1155010000
the reverse of 1155010000: 0000105511
```

Example: Palindrome

- The string remains unchanged after reverse
- Example
  - "racecar"



r	a	c	e	c	a	r

Example: Palindrome

```
Scanner scanner = new Scanner(System.in);
String str = scanner.nextLine();
for(int i=0; i<str.length()/2; i++){
    int lPos = i;   //the left half index
    int rPos = (str.length()-1) - i;  //the right half index
    if(str.charAt(lPos) != str.charAt(rPos)){
        System.out.println(str +" isn't a Palindrome");
        return;
    }
}
System.out.println(str +" is a Palindrome");
```

Once not equal, end comparing and return.

```
tutorial
tutorial isn't a Palindrome
```

Example: HKID check digit

- Every Hong Kong identity card contains

  -one or two alphabets

  -six digits

  -a ***check digit*** in bracket

  e.g.: C689127(4)

- Error checking by check digit

Example: HKID check digit

Check digit computation:

- **Assumption:** a HKID number has the form $ad_1d_2d_3d_4d_5d_6(c)$
  - a is the alphabet
  - Six digits: $d_1d_2d_3d_4d_5d_6$
  - c is the check digit to be computed
- How to compute the check digit?

HKID check digit algorithm

- **Step 1**:Computed the weighted sum S of the HKID number

$$\begin{tikzcd}
a \arrow[d] & d_1 \arrow[d] & d_2 \arrow[d] & d_3 \arrow[d] & d_4 \arrow[d] & d_5 \arrow[d] & d_6 \arrow[d] \\
8 & 7 & 6 & 5 & 4 & 3 & 2
\end{tikzcd}$$

Multiply

$8 \times a$	$7 \times d_1$	$6 \times d_2$	$5 \times d_3$	$4 \times d_4$	$3 \times d_5$	$2 \times d_6$

Sum

$$S = 8 * a \ + 7 * d_1 \ +6 * d_2 \ +5 * \ d_3 + 4 * \ d_4 + 3 * \ d_5 \ +2 * d_6$$

Attention: a is considered as a number which represented the alphabetical order of the alphabet.
That is, ‘A’ is 1,’B’ is 2,’C’ is 3,…,and ‘Z’ is 26

HKID check digit algorithm

- **Step 2**: Find the remainder r of S when divided by 11

  r= S mod 11

- **Step 3**: Subtract remainder the from 11.

  The result is the check digit: c=11-r

- If the subtraction result is 11:
  - Change c to 0;
- If the result is 10:
  - change c to ‘A’.

Example: HKID check digit

Example 1 C689127(?)

1. S = 8 *3 +7* 6+ 6* 8+ 5* 9+ 4* 1+3* 2+2* 7 =183
2. Remainder r =183 mod 11 =7.
3. Finally, c = 11 -7 =4.
4. Thus, the full HKID number with check digit is C689127(4).

Thinking:

How to extract every digit?

How to turn each digit in number for calculation?

Example: HKID check code

```
Scanner scanner = new Scanner(System.in);
String hkid = scanner.nextLine();
int sum = 0;          //to store the weighted sum
int mod;              //the mod of sum % 11
int c;                //11 - mod
int length = hkid.length();  //calculate the length of input
for(int i = 0; i<length; i++){
    if(hkid.charAt(i)>64)      //alphabet digit
        sum = sum + (hkid.charAt(i)-64)*(8-i);    //ascii code 'A' = 65
    else
        sum = sum + (hkid.charAt(i)-48)*(8-i);     //ascii code '0' = 48
}
mod = sum % 11;
c = 11 - mod;
if(c == 11)   c = 0;
if(c == 10)  c = 17;
c = c + 48;              //ascii code '0' = 48
System.out.println("the check digit is: "+ (char)c);
```

Calculate the weighted sum

The end

Thank you!