Féidearthachtaí as Cuimse
Infinite Possibilities

E X A M P L E
character

String class in Java

Object Oriented programming

OLLSCOIL TEICNEOLAÍOCHTA BHAILE ÁTHA CLIATH
TU DUBLIN
TECHNOLOGICAL UNIVERSITY DUBLIN

Why a whole topic on Strings?

- They’re just pieces of text..?

- Because..

- It’s such a common task to want to manipulate or examine strings in any programming language

We saw before:

Primitive Data Types	
int	
short	
long	
byte	
float	
double	
chat	
boolean	

String not listed

A few basics

- String   - is a class (in java)

- String name = “Barnie”;

// creates a reference of type String to an object called name

- If String is a class, why don’t we create Strings like other objects? i.e.

```
String name = new String (“Barnie”)?
```

- Because..

Because

- Java language designers decided that it is so common to create String objects – they'd make it simple

  (.. Like using primitive types)

String class

- **Has a whole set of useful methods for examining or using Strings**
  - E.g. length(), concat(), contains(), equals()
  - Exact methods are in the String class in the Java API

.equals() for comparing strings

- CHECK the String class if you want to do any String manipulation/ checking

Common tasks

- **Programming the following:**

- Check if string a is a subset of string b
  - E.g . is “car” a subset of “escarpment”

- Check if and where a character x occurs in string y

  e.g. (where) does “m” occur in Christmas?

Look up the API

contains

```
public boolean contains(CharSequence s)
```

Returns true if and only if this string contains the specified sequence of char values.

Parameters:

s - the sequence to search for

Returns:

true if this string contains s, false otherwise

Throws:

NullPointerException - if s is null

Since:

1.5

Check if string a is a subset of string b

E.g . is “car” a subset of “escarpment”

Write the code

Look up the API

indexOf

```
public int indexOf(int ch)
```

Returns the index within this string of the first occurrence of the specified character. If a character with value ch occurs in the character sequence represented by this String object, then the index (in Unicode code units) of the first such occurrence is returned. For values of ch in the range from 0 to 0xFFFF (inclusive), this is the smallest value *k* such that: this.charAt(*k*) == ch is true. For other values of ch, it is the smallest value *k* such that: this.codePointAt(*k*) == ch is true. In either case, if no such character occurs in this string, then -1 is returned.
Parameters:ch - a character (Unicode code point).

Returns:the index of the first occurrence of the character in the character sequence represented by this object, or -1 if the character does not occur.

Check if and where a character x occurs in string y

e.g. (where) does “m” occur in Christmas?

Write the code

Some good gatekeepers

1. .trim()
2. .isEmpty() / .length()
3. .equalsIgnoreCase(String other)
4. toLowerCase() / .toUpperCase()
5. .matches(String regex)
6. .charAt(int index) and .indexOf(char)
7. .contains(String str)
8. .substring(int start, int end)
9. .replaceAll(String regex, String replacement)
10. .split(String regex)

int Gatekeepers

Check / Method	What it does	Example in Gatekeeping
Range check	Ensure value is within a safe min/max	if (age < 0
Non-negative	Prevent negative numbers where not allowed	if (stock < 0) { stock = 0; }
Upper bound	Cap very large values	if (points > 1000) { points = 1000;
Even/odd check	Ensure only even or odd numbers	if (n % 2 != 0) { n = n + 1; //force evn

In groups

- First --- in pseudocode– How would you change a String Name value so that the first letter in a Person’s First and Surname Name was capitalised
- Look at the Strings API and pick the methods you would use

Answer

```
public void setName(String inputName) {
   if (inputName == null || inputName.isEmpty()) {
      System.out.println("Invalid name. Please try again.");
      return; // don’t change the instance variable
   } else {
      String firstLetter = inputName.substring(0, 1).toUpperCase();
      String rest = inputName.substring(1).toLowerCase();
      this.name = firstLetter + rest;
   }
}
```

- Person p = new Person();
- p.setName("");   // prints "Invalid name. Please try again."
- System.out.println(p.getName()); // null (still unset)

- p.setName("mARY");
- System.out.println(p.getName()); // Mary