DISTRIBUTED APPLICATIONS
SESSION 3

Eng. Mouhsen Ibrahim

CONTENTS

- Number class is an abstract class used to represent numbers.
- Integer, Float, Short, Byte, Long and Double inherit from it.
- These children classes implement these methods.
  - xxxValue() convert number to xxx data type such as intValue(), floatVaulue() etc…
  - equals() Check equality between a number object and given value.
  - parseXxx() convert string to a number, Integer.parseInt, Float.parseFloat etc…

NUMBERS

- The Character class is used to represent a single character.
- We can assign char literals to Character objects.
- Escape sequences can be used for some special characters such as \n for new line \t for tab etc…
- The most important methods are:
  - Character.isLetter() to determine if the char is a letter or not.
  - Character.isDigit() to determine if the char is a digit or not.
  - Character.isUpperCase() and Character.isLowerCase() to determine the case of the char.
  - Character.toLowerCase() and Character.toUpperCase() to change the case of the char.

CHARACTERS

- Strings are arrays of characters, they are treated as objects.
- Strings can be created using literals or arrays of characters.
- We can find the number of characters in a string using length() method.
- We can concatenate strings using + operator or concat() method.
- We can format strings using String.format() method.
- The charAt() method is used to retrieve a char from a string using index.
- The indexOf() method to find the index of a string or char.
- endsWith() or startsWith() are used to check if the string end or start with a substring.

STRINGS

- toCharArray() convert a string to a character array.
- trim() remove all leading and trailing white space characters.
- subString() return a substring from a string.
- replace() return a new string replacing all old chars with new char.
- getBytes() return a byte array from the string.

STRINGS - CONT

- Arrays are used to represent a fixed number of elements of the same type.
- They are declared as follows: int[] arrs = new int[10];
- arrs.length returns the number of elements in an array.
- Arrays can be accessed using for and foreach loops.
- The Arrays class has static methods to work with arrays.
- Arrays.sort() can be used to sort an array.
- Arrays.binarySearch() can be used to apply binary search algorithm to an array after it is sorted.
- Arrays.fill() is used to fill all array elements with a fixed value.

ARRAYS

- Exceptions are runtime errors, they can be only caught in runtime using try/catch block.
- All exception classes inherit from Exception class, when using multiple catch blocks the most general exceptions must be at last and the specific ones at start.
- The method getMessage() returns a message describing the exception.
- The method printStackTrace() prints a trace of all methods executed until we reach this method and did not return.
- Exceptions can be thrown manually using throw statement.

EXCEPTIONS

- A method that may cause an exception can declare this using throws statement after the closing parameters bracket and before the start of the method’s body.
- The finally block is executed all the time whether an exception happened or not.
- We can create user defined exceptions by extending the Exception class.

EXCEPTIONS - CONT

- Interfaces are used to separate method declaration from implementation.
- Methods are declared in interfaces and are then implemented using classes, this helps to keep code organized.
- A class can implement multiple interfaces.
- We can define static and final attributes in interfaces but not instance attributes.
- An interface can extend multiple interfaces.

INTERFACES

- Packages are used to organize code and similar classes and types together.
- Each package has its own naming space so classes can have the same names in different packages.
- The package statement at the first of the file denotes the package where a file resides.
- The import statements are used to import other packages s they can be used in this file.

PACKAGES

- Define an interface called MathUtilsInt which has these methods
- A method for finding the max value between two input arguments, this method must accept arguments from all numeric types: byte, short, int, long, float, double.
- A method to calculate median and average value of the elements of an input int array, it should throw a user defined exception if array is empty.
- A method to find the sum of elements in two input arrays and return the value as an array.
- Define a class MathUtils that implements the previous interface and all its methods.

EXERCISE

THE END