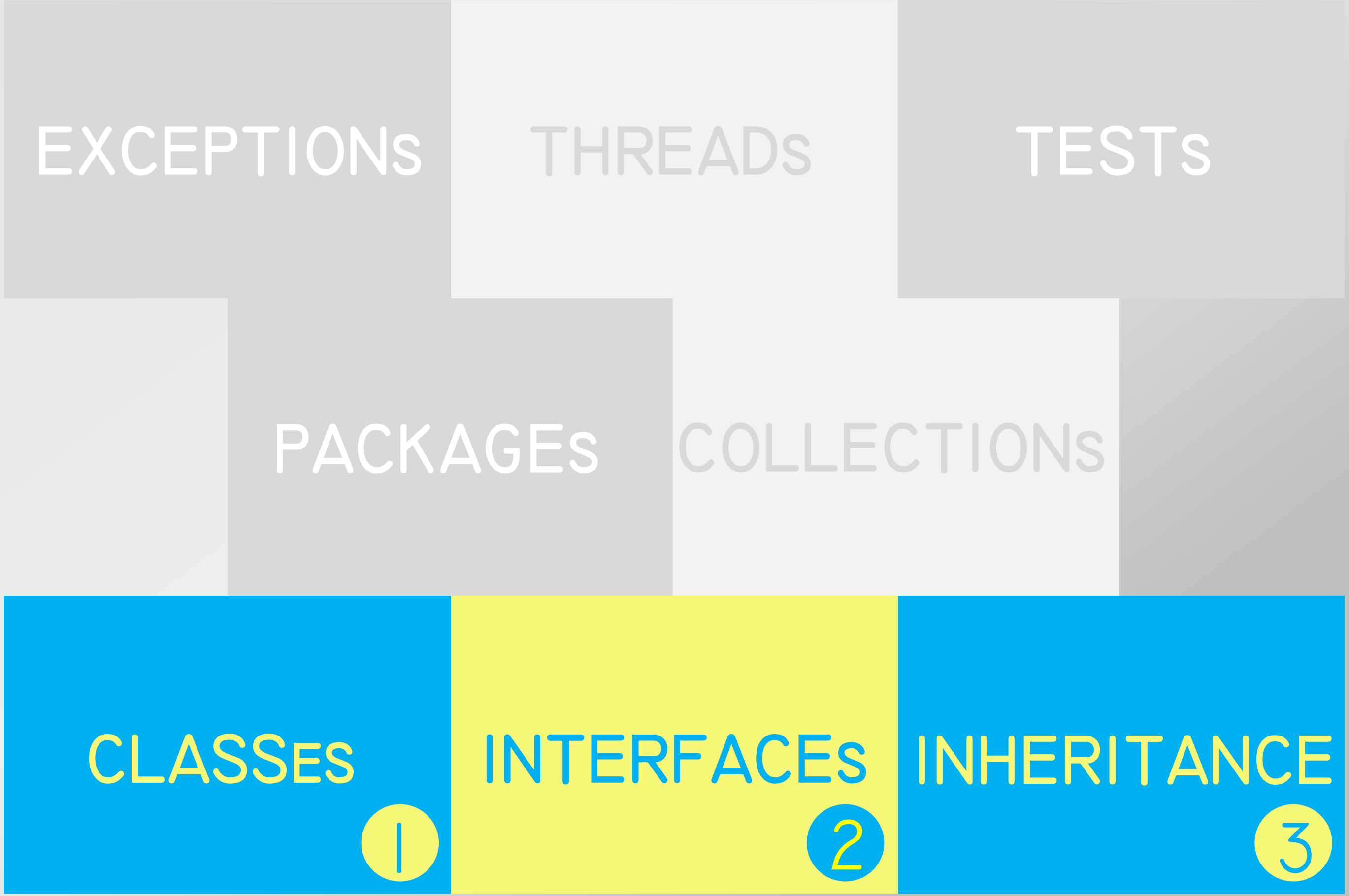
BASICS of Java



EXCEPTIONs

THREADs

TESTs

PACKAGEs

COLLECTIONs

CLASSEs 1

INTERFACEs 2

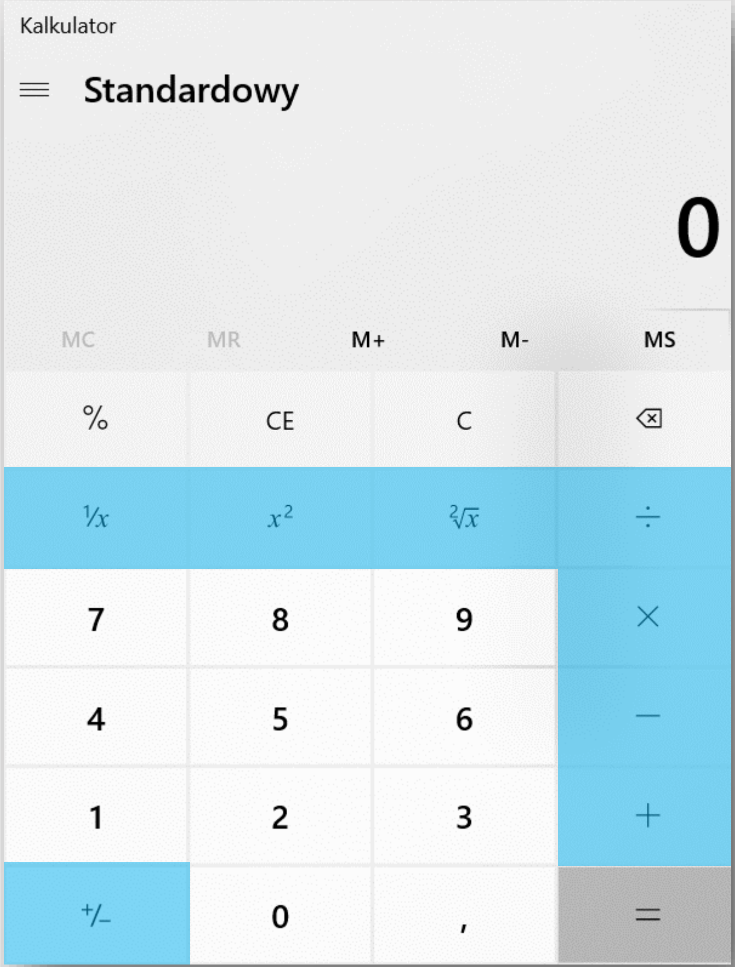
INHERITANCE 3

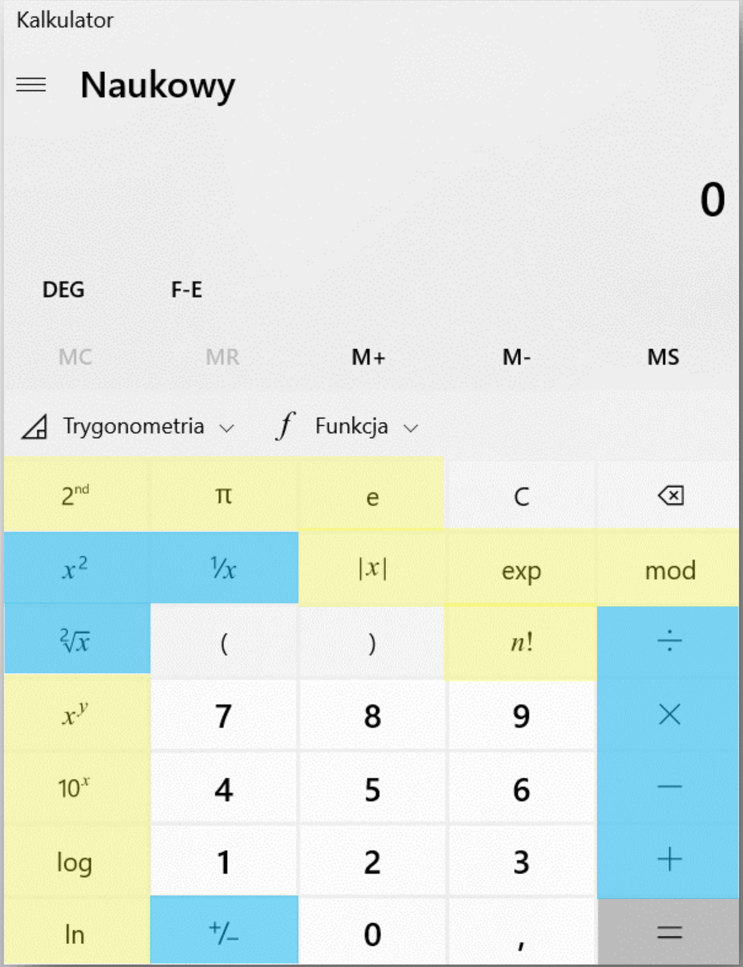
Inheritance

How to creare a subclass (= child class)

How to use methods of the super class (= parent class)

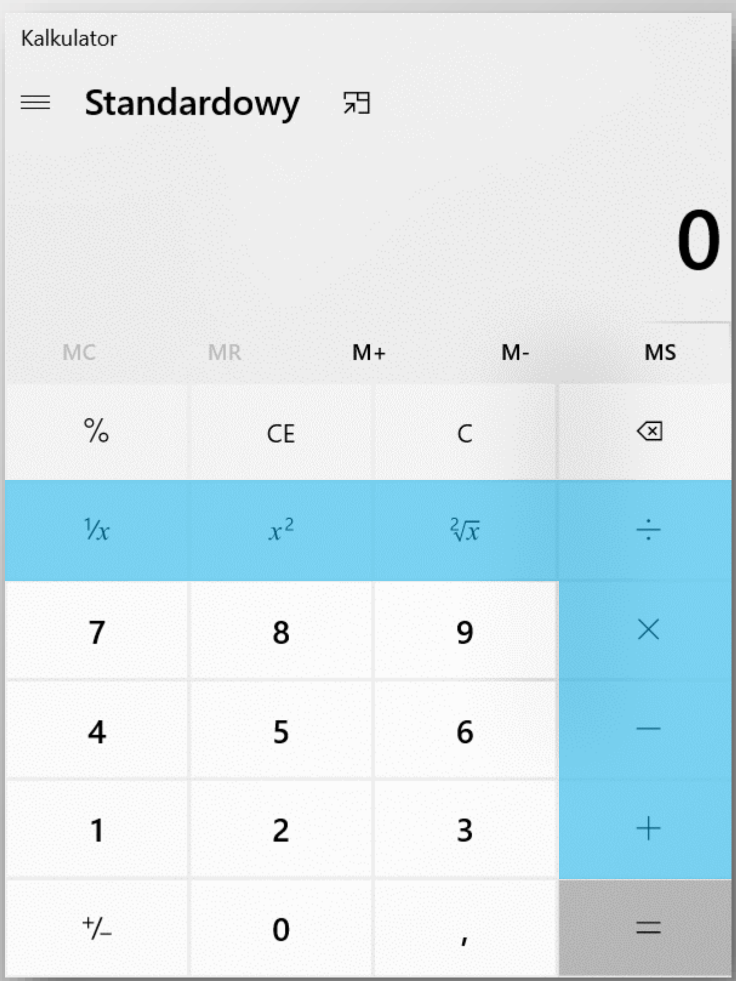
static method


Kalkulator
Standardowy
0
MC
MR
M+
M-
MS
%
CE
C
⌫
$\frac{1}{x}$
x^2
$\sqrt[2]{x}$
÷
7
8
9
×
4
5
6
−
1
2
3
+
+/−
0
,
=


Kalkulator
Naukowy
0
DEG
F-E
MC
MR
M+
M-
MS
Trygonometria
f Funkcja
2^{nd}
π
e
C
⌫
x^2
$\frac{1}{x}$
$|x|$
exp
mod
$\sqrt[2]{x}$
(
)
$n!$
÷
x^y
7
8
9
×
10^x
4
5
6
−
log
1
2
3
+
ln
+/−
0
,
=

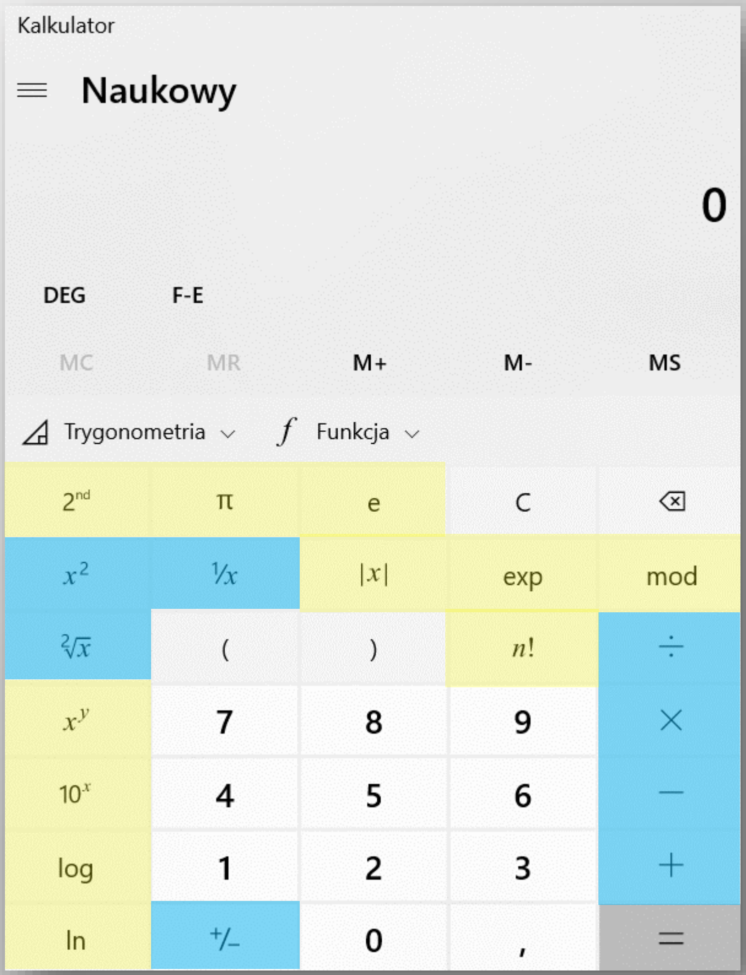
Standard

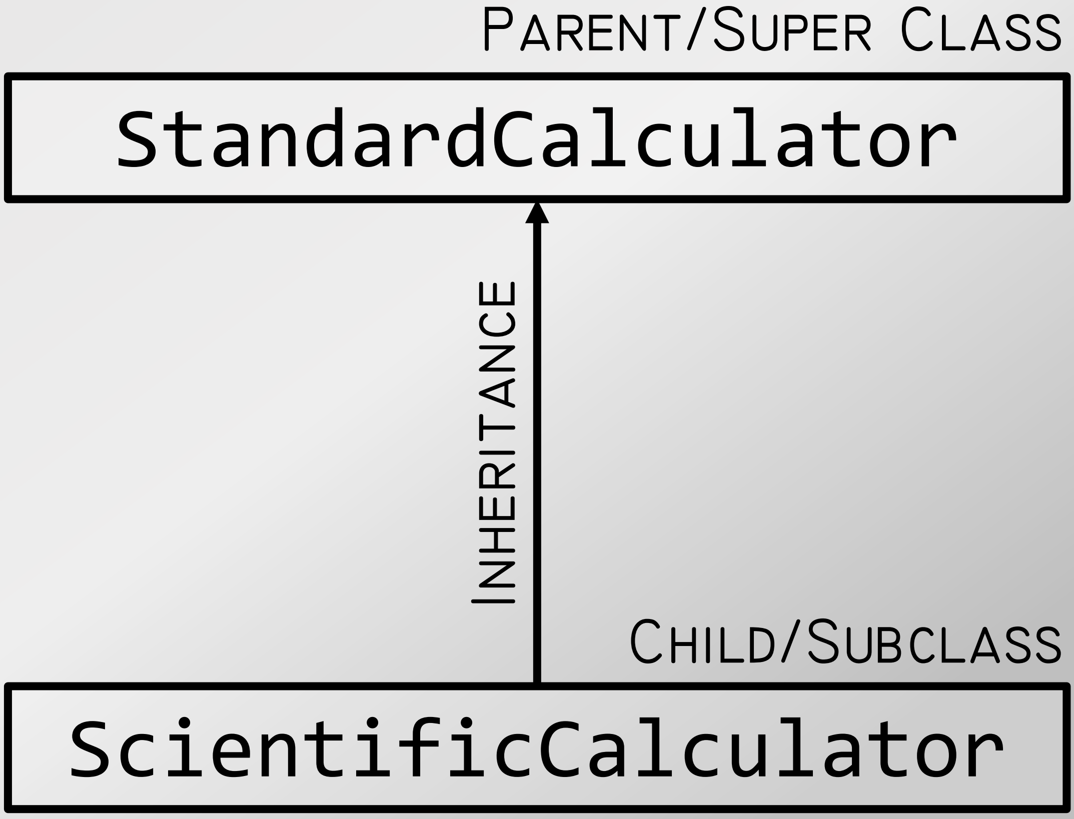
- addition (+)
- substruction (-)
- multiplication (x)
- division (⁒ and 1/x)
- square root ($\sqrt[2]{x}$)
- square (x^2)

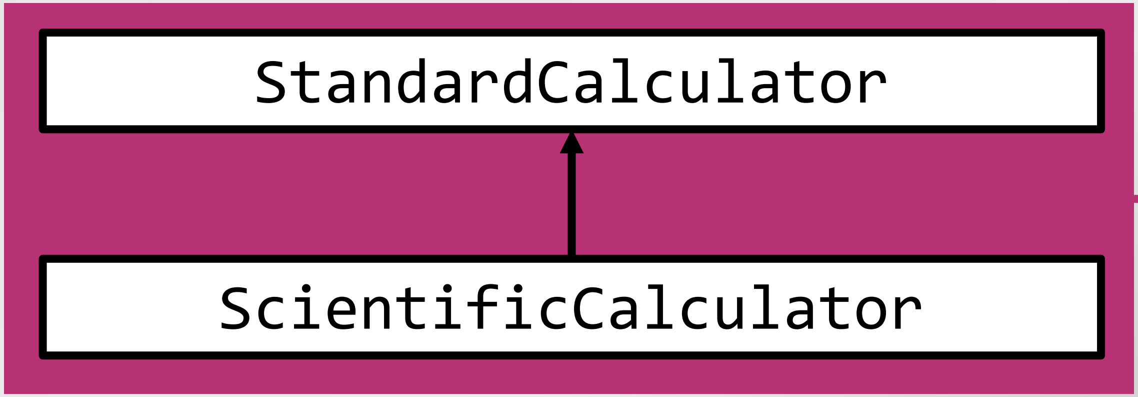


Scientific

- addition (+)
- substruction (-)
- multiplication (x)
- division (÷ and 1/x)
- square root ($\sqrt[2]{x}$)
- square (x^2)
- natural logarithm (ln)
- exponent (exp)
- …




Parent/Super Class
StandardCalculator
Inheritance
Child/Subclass
ScientificCalculator



StandardCalculator

ScientificCalculator

```
public class StandardCalculator {
}
```

```
public class ScientificCalculator extends
KalkulatorStandartowy {
}
```

1-5

□ CREATE A CLASS CALLED <STANDARDCALCULATOR>

□ ADD TWO METHODS add() AND substruct()

□ CREATE A CLASS CALLED <SCIENTIFICCALCULATOR> THAT IS A SUBCLASS OF THE <STANDARDCALCULATOR> CLASS (USE THE KEYWORD **extends**)

□ TO THE <SCIENTIFICCALCULATOR> CLASS ADD TWO METHODS modulo() AND factorial()

REMAINDER OPERATOR (%)

$$n! = n \times (n-1) \times (n-2) \times ... \times 2 \times 1$$

Ex., $5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$

□ IN THE MAIN PROGRAMM ASK A USER TO ENTER A INTEGER NUMBER, CREATE AN OBJECT OF THE <CalculatorScientific> CLASS, AND DISPLAY THE RESULT OF THE factorial() METHOD. NEXT, ASK A USER TO ENTER ANOTHER INTEGER NUMBER. DISPLAY THE RESULTS OF THE add(), substruct(), AND modulo() METHODS.

6-10

□ CREATE A CLASS CALLED <CALCULATORPROGRAMMER> THAT IS A SUBCLASS OF <CALCULATORSTANDARD>

□ ADD TWO METHODS **and()** AND **or()**THAT TAKE AS PARAMETERS TWO INTEGER VARIABLES AND RETURN THE RESULT OF THE LOGICAL AND (&) AND OR (|) OPERATORS.

□ IN THE MAIN PROGRAMM ASK A USER TO ENTER TWO INTEGER NUMBERS, CREATE AN OBJECT OF THE <CALCULATORPROGRAMMER> CLASS, AND DISPLAY IN THE CONSOLE THE RESULT OF THE **and()** AND **or()** METHODS.

□ CHANGE THE ACCESS SPECIFIER OF THE **and()** AND **or()** METHODS FROM **public** TO **static**.

□ UPDATE YOUR MAIN PROGRAM CORRESPONDINGLY.

RETURN VALUE

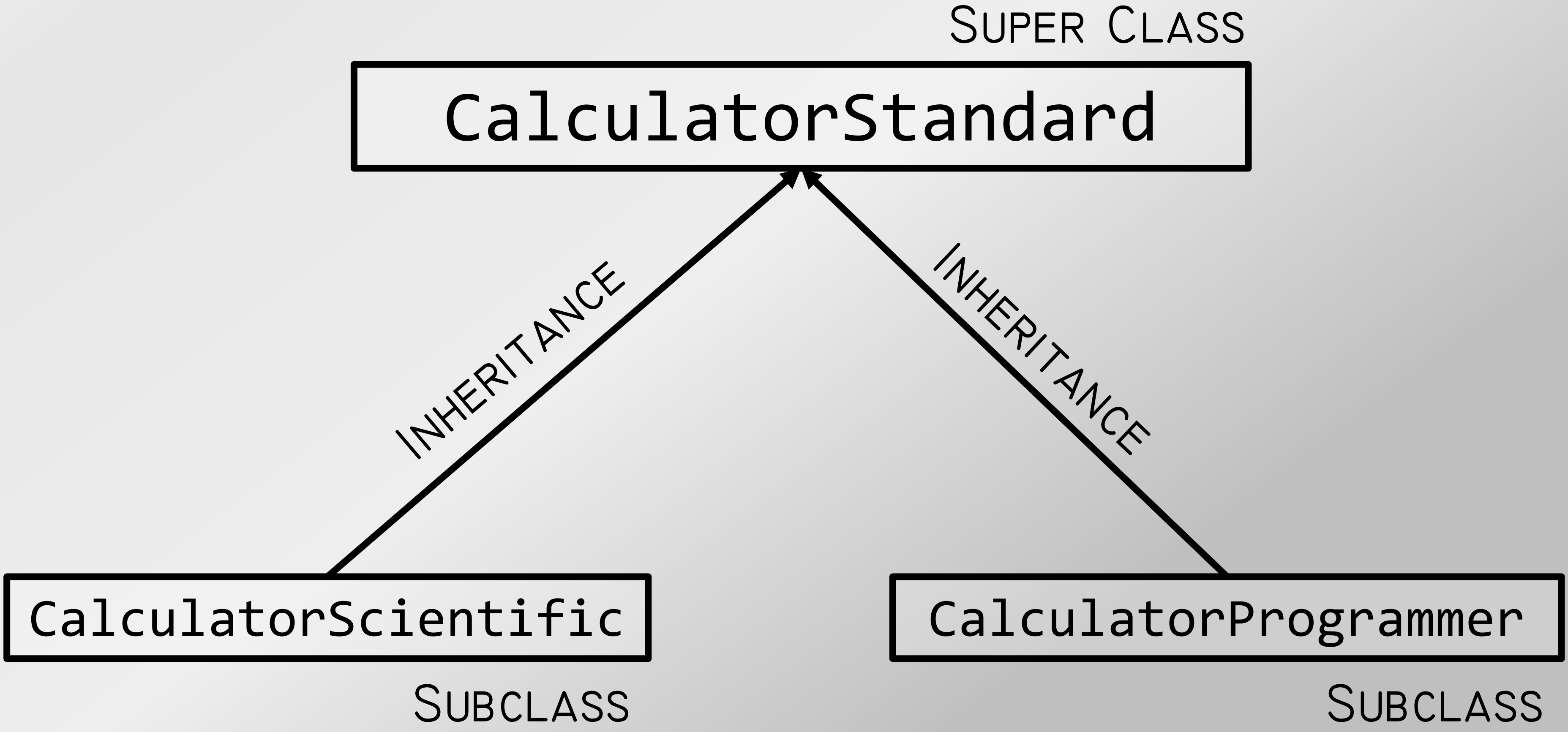
BODY

```
public int methodName(){}
```

ACCESS SPECIFIER

PARAMETERS

(!) THE STATIC KEYWORD



Super Class

CalculatorStandard

Inheritance

Inheritance

CalculatorScientific

Subclass

CalculatorProgrammer

Subclass

- I am sure that you know it, but just in case :P

	AND	
&	0 1 1 1	7
	0 1 0 1	5
	0 1 0 1	5

	OR	
|	0 1 1 1	7
	0 1 0 1	5
	0 1 1 1	7

11-13

□ Create a class called <ItalianBasic>. This class contains two fields of the String Array datatype:
<basicPL> {"cześć","dziękuję","tak","nie"} and
<basicIT> {"ciao","grazie","si","no"}.

□ Add two methods **plTOit()** and **itTOpl()**. The **plTOit()** method takes as an argument a word in Polish (String). The method checks if the <basicPL> string array contains this word and returns its Italian version from the <basicIT> String array. If there is no such word in this array, the method returns "There is no such word in the dictionary".

□ implement the **itTOpl()** method that translates a word from Italian to Polish.

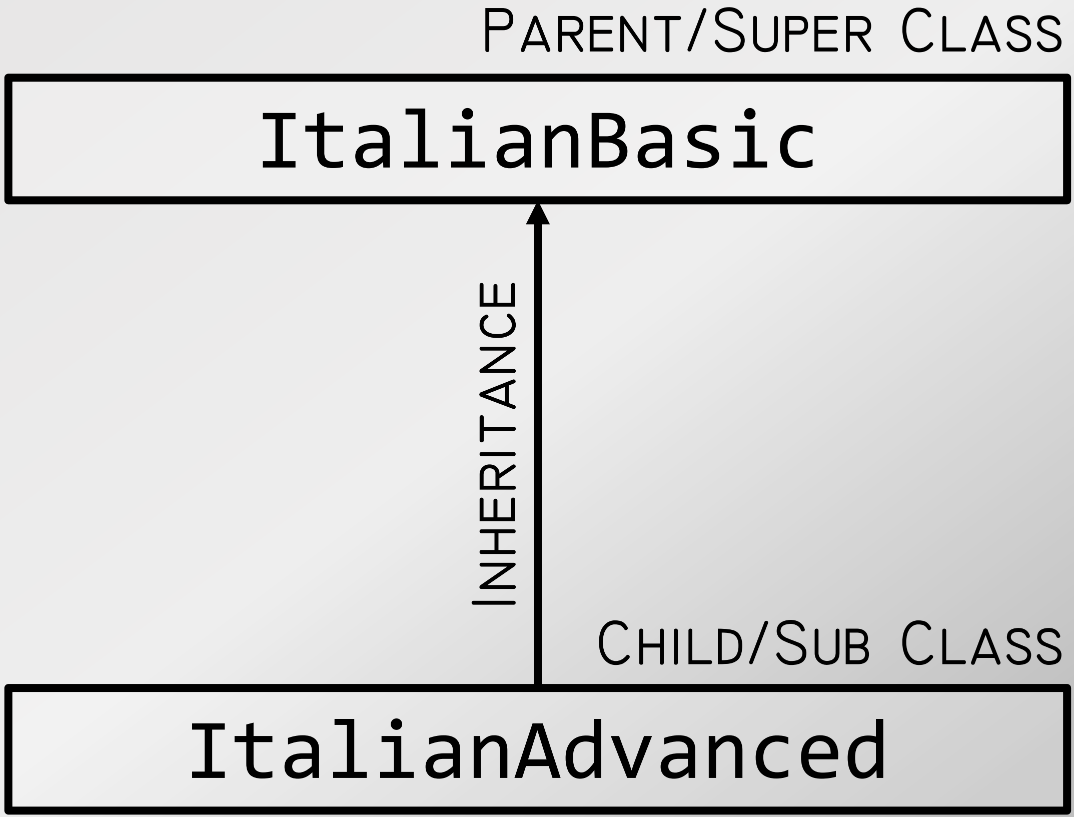
14-17

□ In the main program ask a user to enter a word in Italian and then display its translation in Polish using the `itTOpl()` method.

□ Next, ask the user to enter a word in Polish and then display its translation in Italian using the `plTOit()` method.

□ Change the access specifier of the `itTOpl()` and `plTOit()` methods from **`public`** to **`static`**.

□ Update the rest of the code correspondingly.



Parent/Super Class

ItalianBasic

Inheritance

Child/Sub Class

ItalianAdvanced

! Take a look at this video: ***Polish&Italian.mp4***

□ CREATE A CLASS CALLED <ITALIANADVANCED>. THIS CLASS EXTENDS THE <ITALIANBASIC> CLASS. APART FROM THE <BASICPL> AND <BASICIT> STRING ARRAYS IT HAS TWO ARRAYS WITH NAMES OF NUMBERS (FROM ONE TO THREE) IN POLISH AND IN ITALIAN. ! Take a look at this video: ***Polish&Italian.mp4***

18-19

□ ADD TWO **static** METHODS noTOpl() AND noTOit() THAT TRANSLATE NAMES OF NUMBERS FROM ITALIAN TO POLISH AND VICE VERSA.

20-23

□ In the main program ask a user to enter a word in Italian and then display its translation in Polish (this time using the <ItalianAdvanced> class).

□ Next, ask the user to enter a word in Polish and then display its translation in Italian (again using the <ItalianAdvanced> class).

□ Then, ask the user to enter a name of some number in Polish and display its translation in Italian.

□ In the end, ask the user to enter a name of some number in Italian and return its translation in Polish.

BASICS of Java

EXCEPTIONs THREADs TESTs

PACKAGEs COLLECTIONs

☑ ☑ ☑