

# LAB7 - Expression (III)

LAB of COMP2021 OBJECT-ORIENTED PROGRAMMING

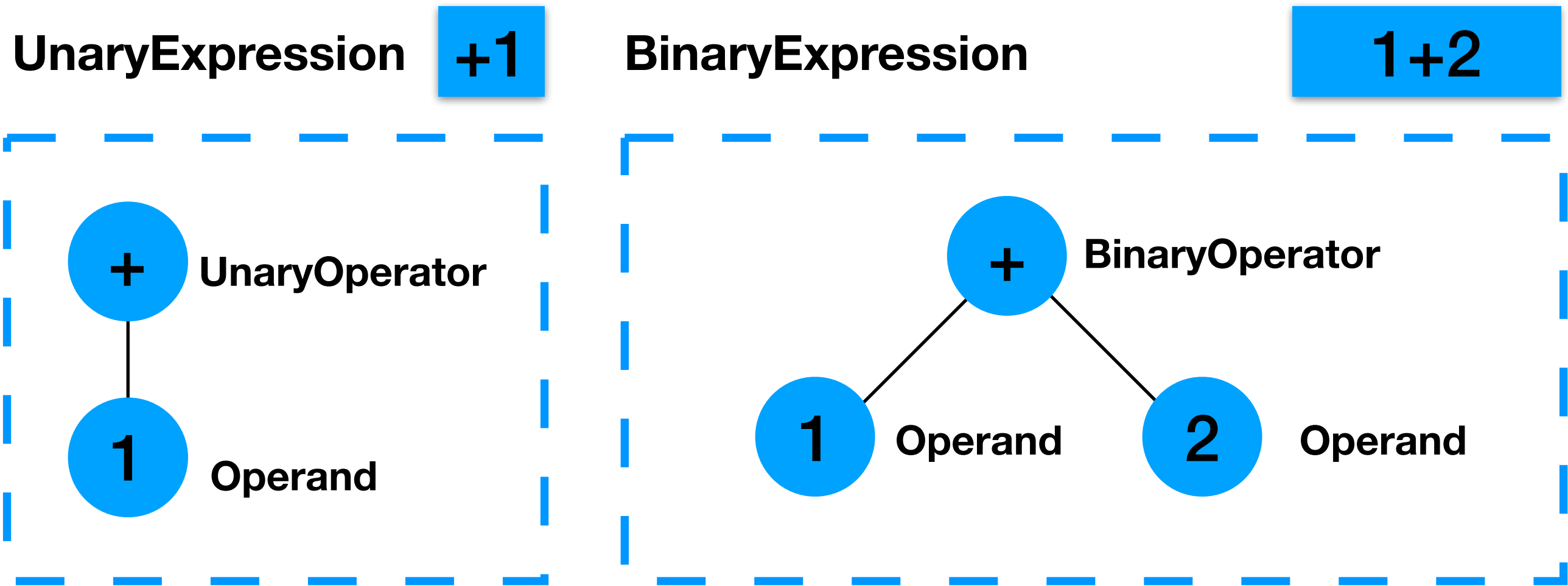
# Objectives

- To review the concepts:
  - Inheritance
  - Abstract Class

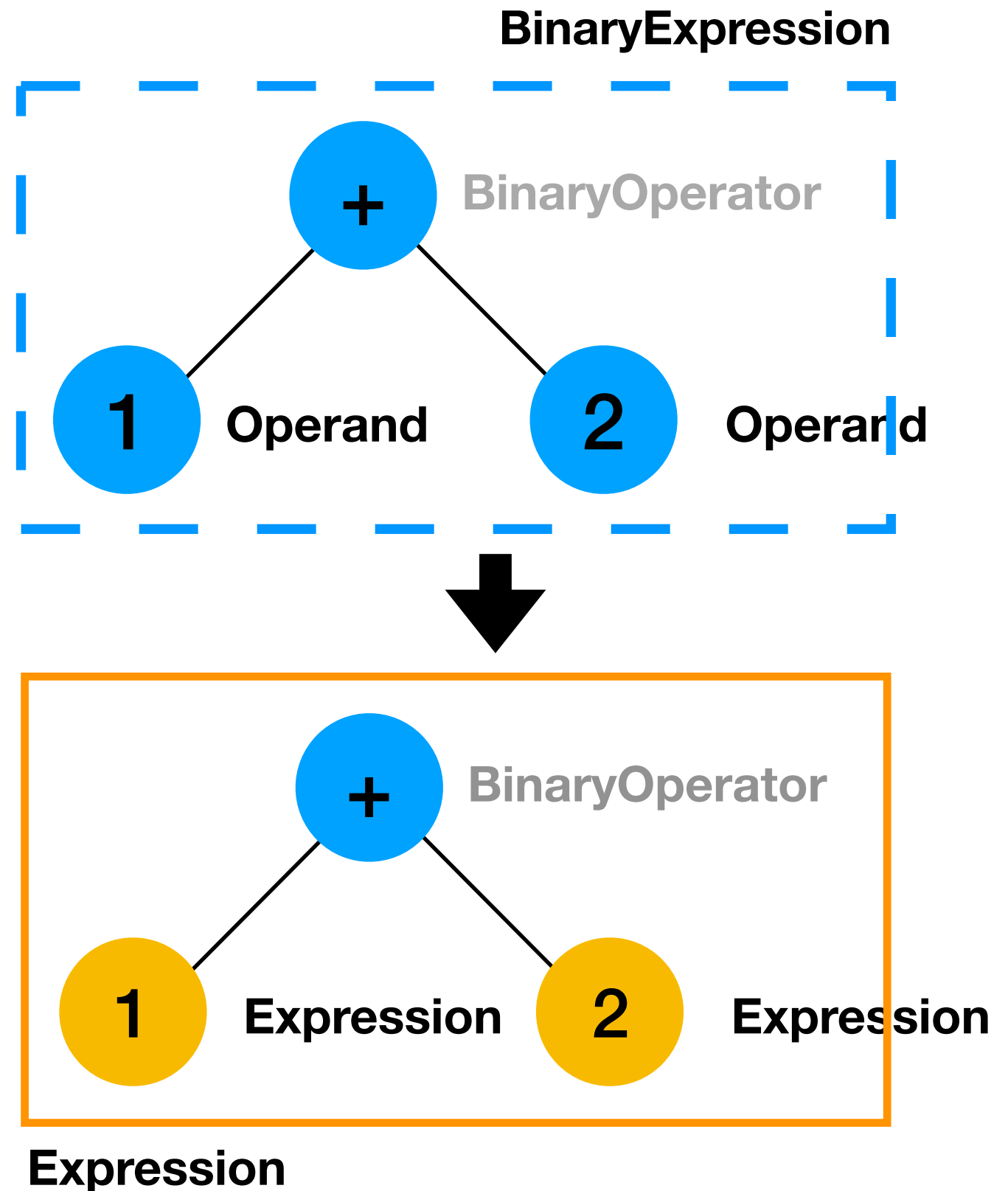
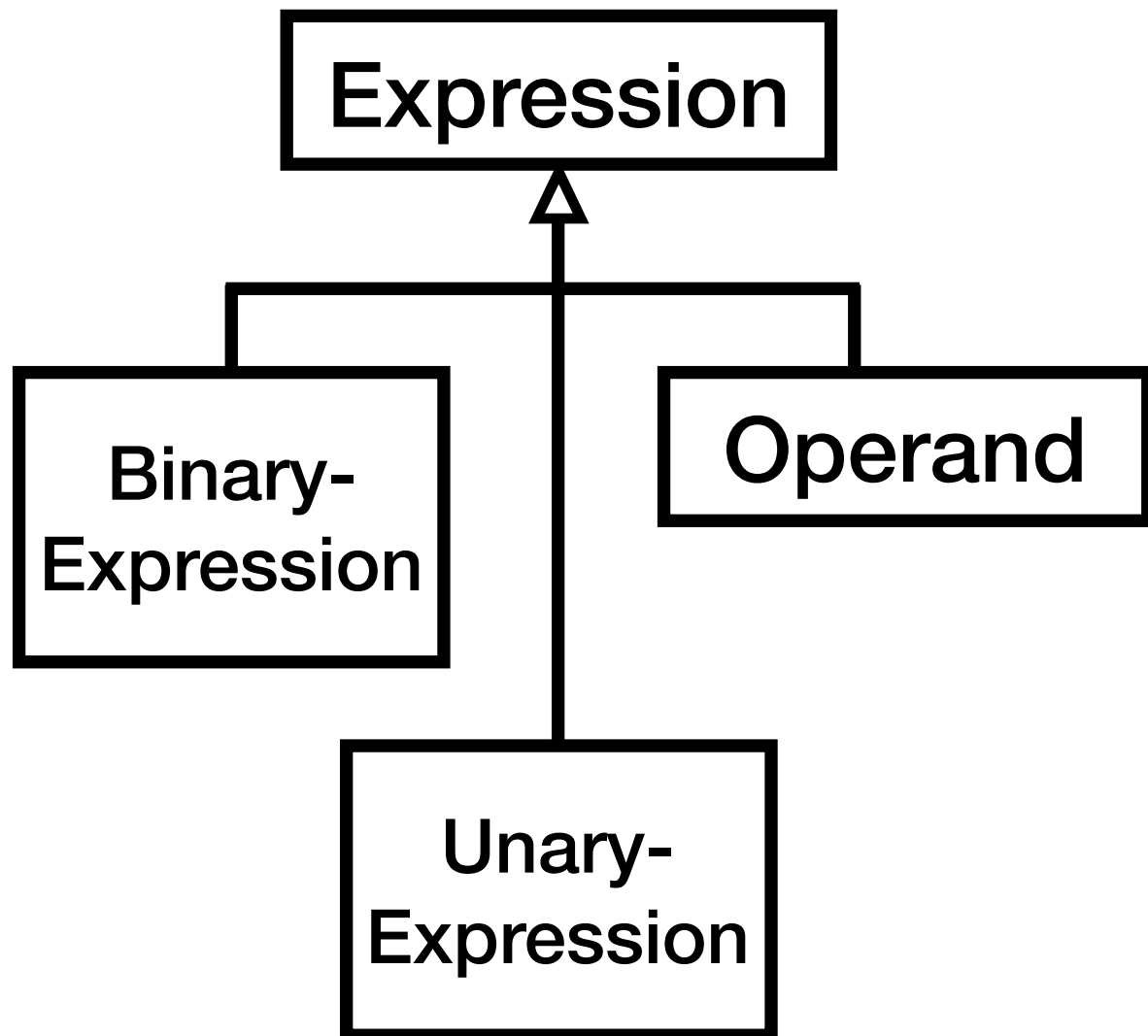
# Inheritance

- Inheritance is an important way to reuse existing code
- Defining a new class by inheriting members from an existing class, then
  - Adding new members and/or
  - Overriding inherited methods
- A superclass defines a more general concept, while a subclass defines a more specific concept

# Current Design

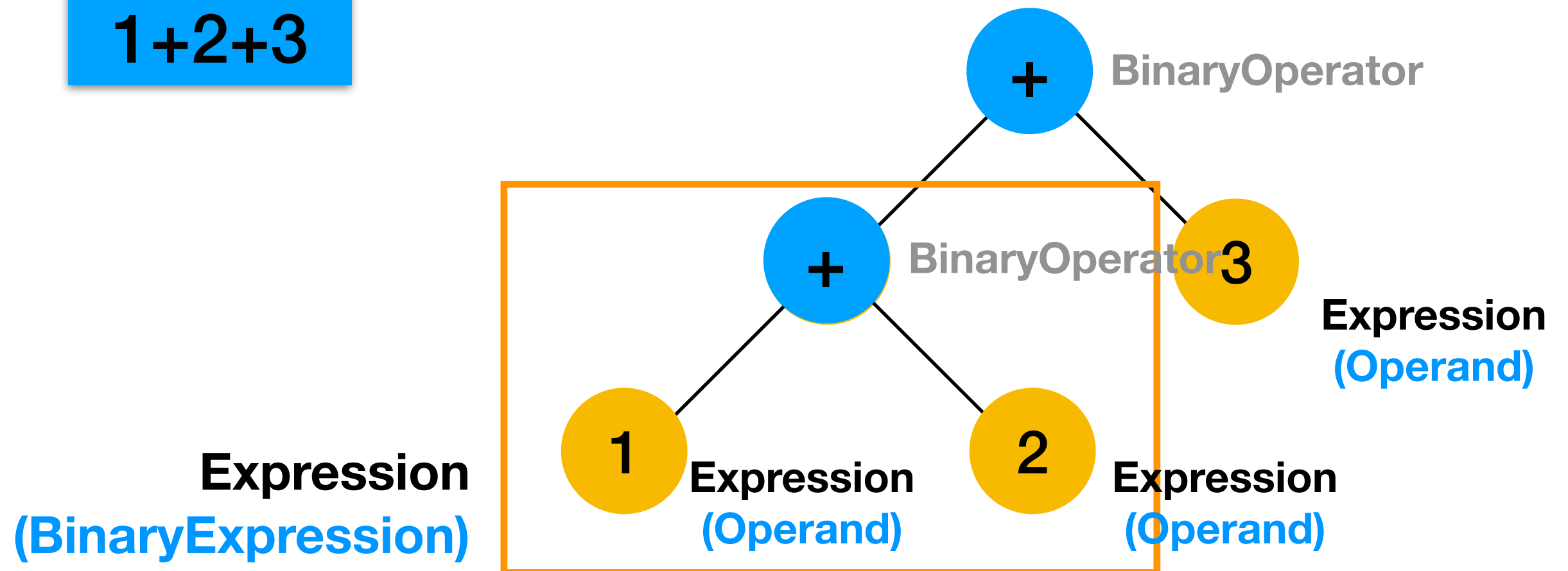


# A more Genetic Design



# A Nested Expression

1+2+3



# Requirement

- Abstract Class Expression
- Operand extends Expression
- BinaryOperator
- BinaryExpression extends Expression
- UnaryOperator

**We use “#” for PLUS and “~” for MINUS to distinguish between BinaryOperator and UnaryOperator.**

- UnaryExpression extends Expression

# Abstract Class

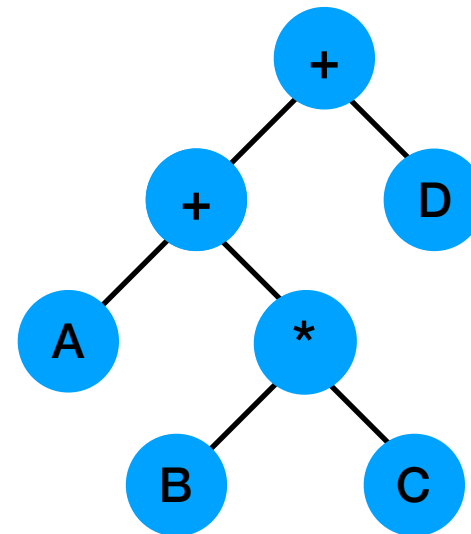
- Keyword: abstract

```
public abstract class Expression {  
    public abstract int evaluate();  
  
    public abstract String toInfixString();  
    public abstract String toPrefixString();  
    public abstract String toPostfixString();  
}
```



# Prefix and Infix Notation

**A+B\*C+D**



Infix Notation	Prefix Notation	Postfix Notation
A + B * C + D	+ + A * B C D	A B C * + D +
(A + B) * (C + D)	* + A B + C D	A B + C D + *
A * B + C * D	+ * A B * C D	A B * C D * +
A + B + C + D	+ + + A B C D	A B + C + D +

**\*To simplify our code, we create “()” for every BinaryExpression in Infix Notation**

- Thank you