



---

Design  
by  
Contract™

# Design by Contract

---



A discipline of analysis, design, implementation,  
management

# Applications

---



Getting the software right

Analysis

Design

Implementation

Debugging

Testing

Management

Maintenance

Documentation



Every software element is intended to satisfy a certain goal, for the benefit of other software elements (and ultimately of human users).

This goal is the element's **contract**.

The contract of any software element should be

- Explicit.
- Part of the software element itself.

# The imperative and the applicative



<b>do</b> <i>balance</i> := <i>balance</i> - <i>sum</i>	<b>ensure</b> <i>balance</i> = <b>old</b> <i>balance</i> - <i>sum</i>
<b>PRESCRIPTIVE</b>	<b>DESCRIPTIVE</b>
How?	What?
Operational	Denotational
<b>Implementation</b>	<b>Specification</b>
Command	Query
Instruction	Expression
Imperative	Applicative

# How not to do it

---



*r*(*i*: *INTEGER*): *BOOLEAN* is

require

$i \geq 0$  or  $i < 0$

ensure

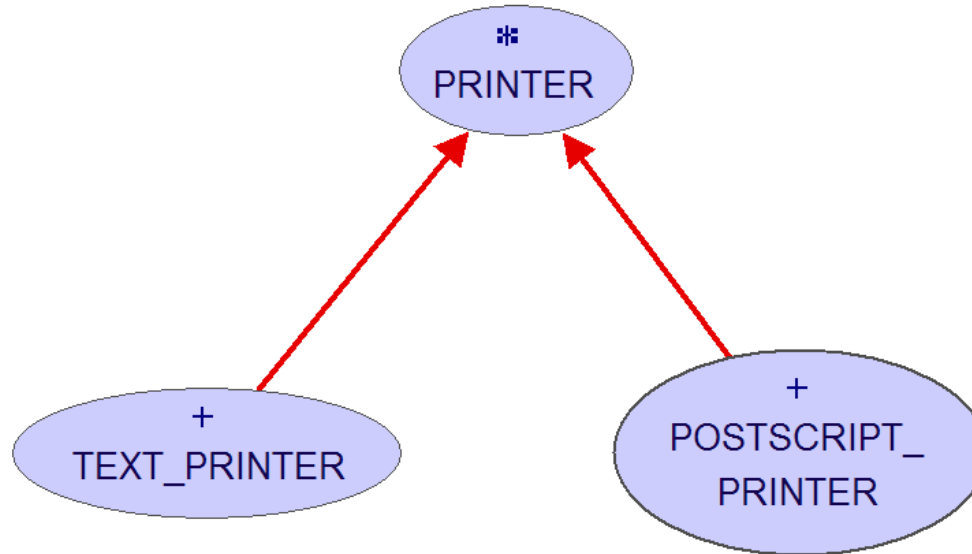
*Result* = true or *Result* = false


*r*(*x*: *BANK\_ACCOUNT*): *BOOLEAN* is

require

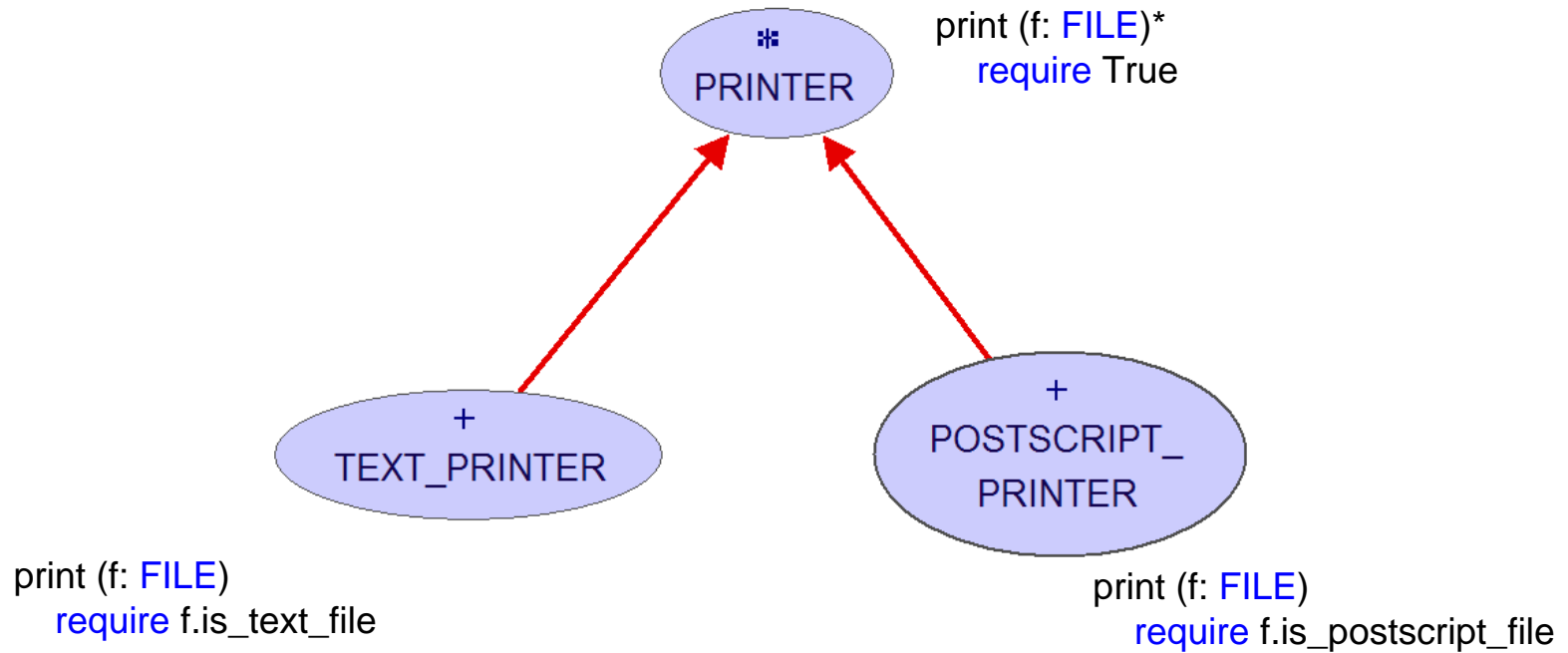
$x \neq \text{Void}$  and  $x.\text{balance} > 0$

# Exercise 2



 Inheritance

# Solution



Is the solution correct?

↑ Inheritance



Issues: what happens, under inheritance, to

- Class invariants?
- Routine preconditions and postconditions?



## Invariant Inheritance rule:

- The invariant of a class automatically includes the invariant clauses from all its parents, "and"-ed.

Accumulated result visible in flat and interface forms.



When redeclaring a routine, we may only:

- Keep or weaken the precondition
- Keep or strengthen the postcondition



# Assertion redeclaration rule in Eiffel

---

A simple language rule does the trick!

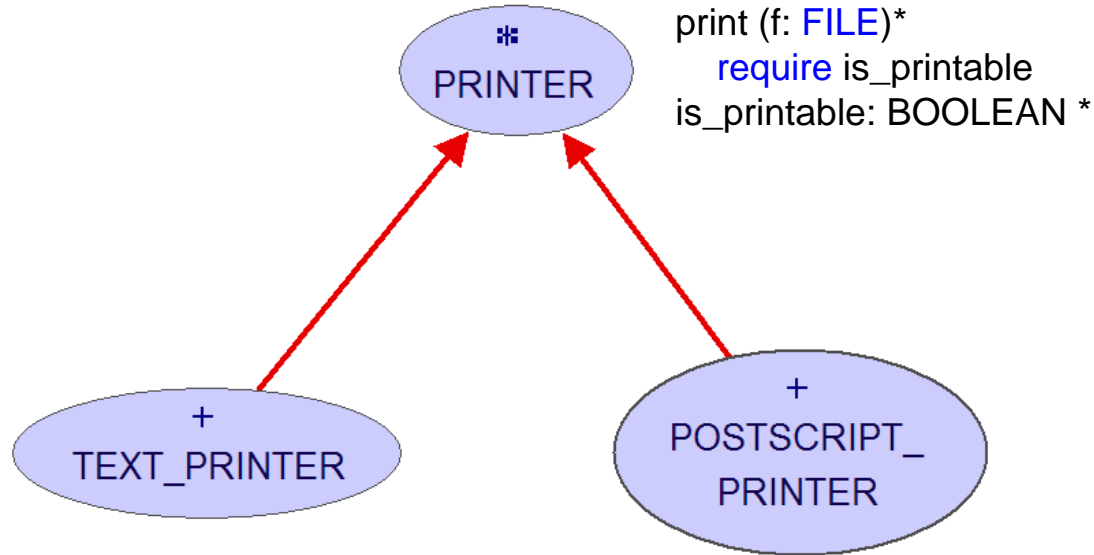
Redefined version may have nothing (assertions kept by default), or

```
require else new_pre  
ensure then new_post
```

Resulting assertions are:

- *original\_precondition* **or** *new\_pre*
- *original\_postcondition* **and** *new\_post*

# Correct Solution



print (f: FILE)\*  
require is\_printable  
is\_printable: BOOLEAN \*

```
is_printable: BOOLEAN  
do  
  Result := f.is_text_file  
end
```

```
is_printable: BOOLEAN  
do  
  Result := f.is_postscript_file  
end
```

 Inheritance