

# Programmation Orientée Objet : Templates



Jean-Cédric Chappelier

Faculté I&C

# Organisation du travail (semestre)

	<b>MOOC</b>	<b>déc.</b>	<b>cours</b> 1 h Jeudi 8-9	<b>exercices</b> 2 h Jeudi 9-11
1	19.02.26		0	Intro + compil. séparée
2	26.02.26	1. Intro POO	0	Intro POO
3	05.03.26	2. Constructeurs/Desi	0	Constructeurs
4	12.03.26	3. Surcharge des opé	0	Surcharge
5	19.03.26	4. Héritage	0	Héritage
6	26.03.26	5. Polymorphisme	0	Polymorphisme 1
7	02.04.26		1	Polymorphisme 2 / Collections hétérogènes
-	09.04.26		-	vacances Pâques
8	16.04.26	6. Héritage multiple	1	Héritage multiple
9	23.04.26		-	Midtem
10	30.04.26	(7. Etude de cas)	-	Templates
11	07.05.26		-	Structure de données abstraites ; Bibliothèques
12	14.05.26		-	(Ascension)
13	21.05.26	(7. Etude de cas)	-	Bibliothèques (fin) + Révisions
14	28.05.26		-	Examen

# Concepts fondamentaux

- ▶ bien comprendre de quoi il s'agit :  
modèles de programmes  
(« infinité potentielle de codes »)
- ▶ modèles dans le(s) type(s) et/ou dans des entiers
- ▶ instanciations implicites et explicites
- ▶ spécialisation de modèles

vrai code C++

```
template < type >  
codeA
```

instan ..

```
codeA < int >  
{ x < int >  
}
```

```
codeA < Particle >  
{ y  
}
```

abstractions

```
codeA < int >  
{ x }
```

```
codeA < double >  
{ x }
```

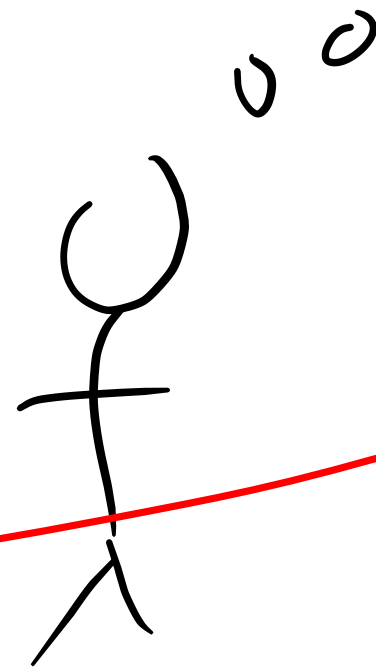
Spécialisation

```
codeA < Particle >  
{ y }
```

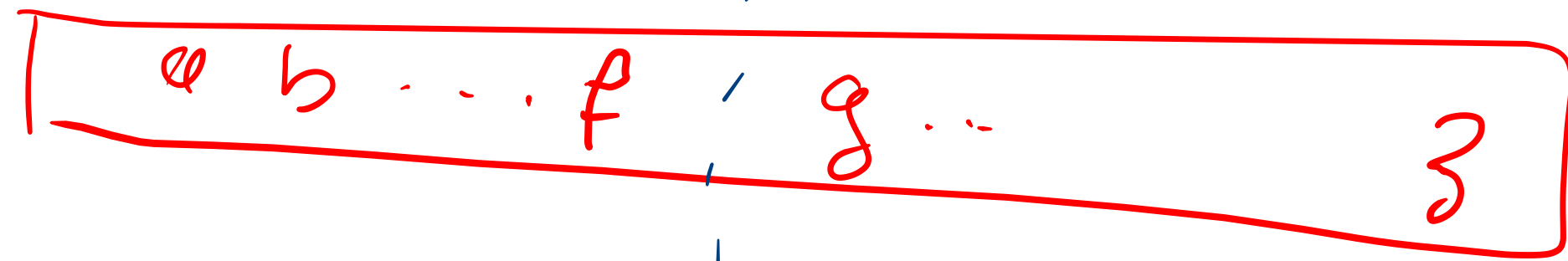
```
codeA < vector < double >  
... { x }
```

A

A

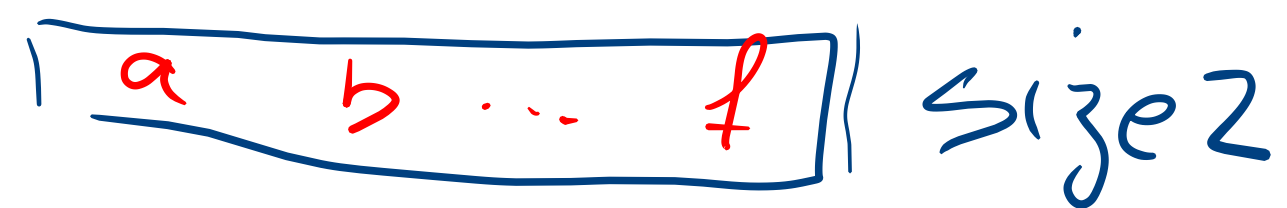


v



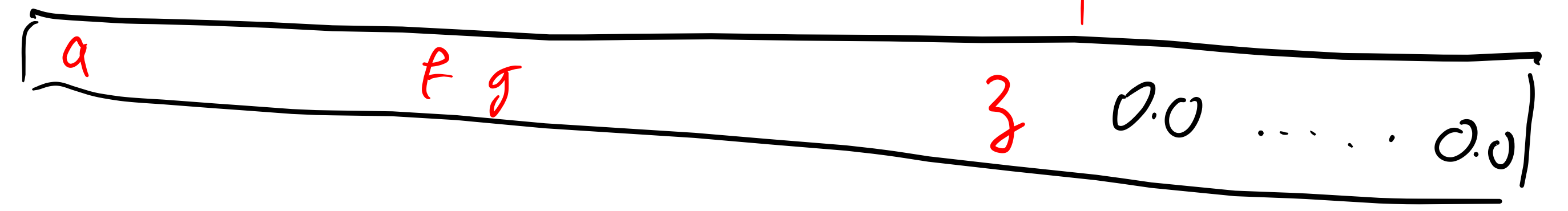
size

v.convert()



size2

v.convert()



size2

# Etude de cas

- ▶ vecteurs (mathématiques) de dimension finie sur un corps :
  - ☞ template en :
    - ▶ la dimension *→ entier*
    - ▶ le corps, double par défaut

*size\_t*

*↳ type*

## Intérêts :

- ▶ template de classe
- ▶ template de fonctions (surcharges externes)
- ▶ template méthode dans le template de classe  
(`convert()` pour changer de dimension)
- ▶ problème de conversion/instanciation implicite lorsqu'ambigüe