

**PROGRAMMATION
CONCURRENTTE**



IV ÉCHAPPEMENTS & EXCEPTIONS

Fabrice.Kordon@lip6.fr



16 - ÉCHAPPEMENT

Programmation Concurrante - LI330
Université P. & M. Curie - année scolaire 2013/2014

PrC

● Mécanisme de déroutement

- Associé à une «condition de levée» (raise condition)
- Association à un «traitement» (handler)

● Déroutement «propre»

- Un échappement préserve le contexte du programme au niveau du traitement associé



● Vision «de haut niveau» des interruptions

- On peut «empiler» les échappements

On peut «blinder» un programme sans échappements

- Cela alourdi considérablement l'écriture
- Cela altère la lisibilité
- Cela peut nuire à l'évolutivité
- Incommode de prévoir un «code d'erreur»
 - Cela alourdit considérablement le code
 - Comment faire dans certains cas
 - **Exemple:** l'addition entre 2 entiers peut générer une erreur (dépassement de capacité) quelle valeur associer à ce cas de figure?

Traitement des situations exceptionnelles

- Impossible à intégrer dans les sous-programmes
 - Comment savoir ce que veut faire «l'appelant»?
- D'où la règle
 - Toujours différer les traitements dépendants de l'appelant

①

Les échappements permettent de factoriser le traitement de cas exceptionnels (e.g. erreurs)

②

C'est un mécanisme qui préserve la lisibilité des programmes

③

**Ada, Java, C++ sont dotés d'un mécanisme d'exceptions
D'autres langages (comme C) ne le sont pas.**