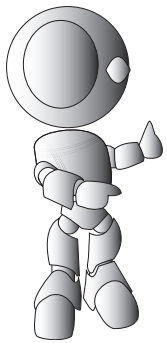
Tests

Cours de génie logiciel

Tuyêt Trâm DANG NGOC
<dntt@u-cergy.fr>

Université de Cergy-Pontoise

2012–2013



DÉPARTEMENT
Sciences Informatiques

Plan

DÉPARTEMENT
Sciences Informatiques

DÉPARTEMENT
Sciences Informatiques

Pourquoi des tests ?

- Tout programme est soit trivial soit contient au moins un bug.
- Loi de Murphy : si une catastrophe peut se produire alors elle se produira
- Mieux vaut trouver les bugs lors du développement plutôt qu'attendre que le client (ou le prof) le trouve.

But des tests : limiter la casse !

Fatalité du Programmeur

Tout programme est soit trivial soit contient au moins un bug.

- Algorithme mal conçu ou mal implémenté
- Mauvaise gestion de la mémoire (fuite, mauvais adressage, etc.)
- Mauvaise gestion de la concurrence (synchronisation, ordonnancement, etc.)
- Valeurs non ou mal initialisées
- la faute aux bibliothèques, au compilateur, au système, à l'ordinateur...
- ...

Tests de fuite mémoire

Identification des problèmes mémoire

- mémoire non libéré
- accès à un espace mémoire non réservé

Outils Pour C : purify

```
int main(void) {
    char *chaine1 = malloc(sizeof(char)*50);
    char *chaine2 = malloc(sizeof(char)*50);
    scanf("%s", string2);
    chaine1 = chaine2;  /* espace mémoire de chaine1 perdu */
    free(chaine2);
    free(chaine1); /* Erreur, espace déjà libéré*/
    return 0;
}
```

DÉPARTEMENT Sciences Informatiques

Tests de cas d'utilisation (use-case)

- Batterie de tests couvrant un éventail de cas possibles est appliqué.
- Cas extrêmes
- Cas moyens

Tests de couverture

Loi de Murphy

Si une catastrophe peut se produire, alors elle se produira !

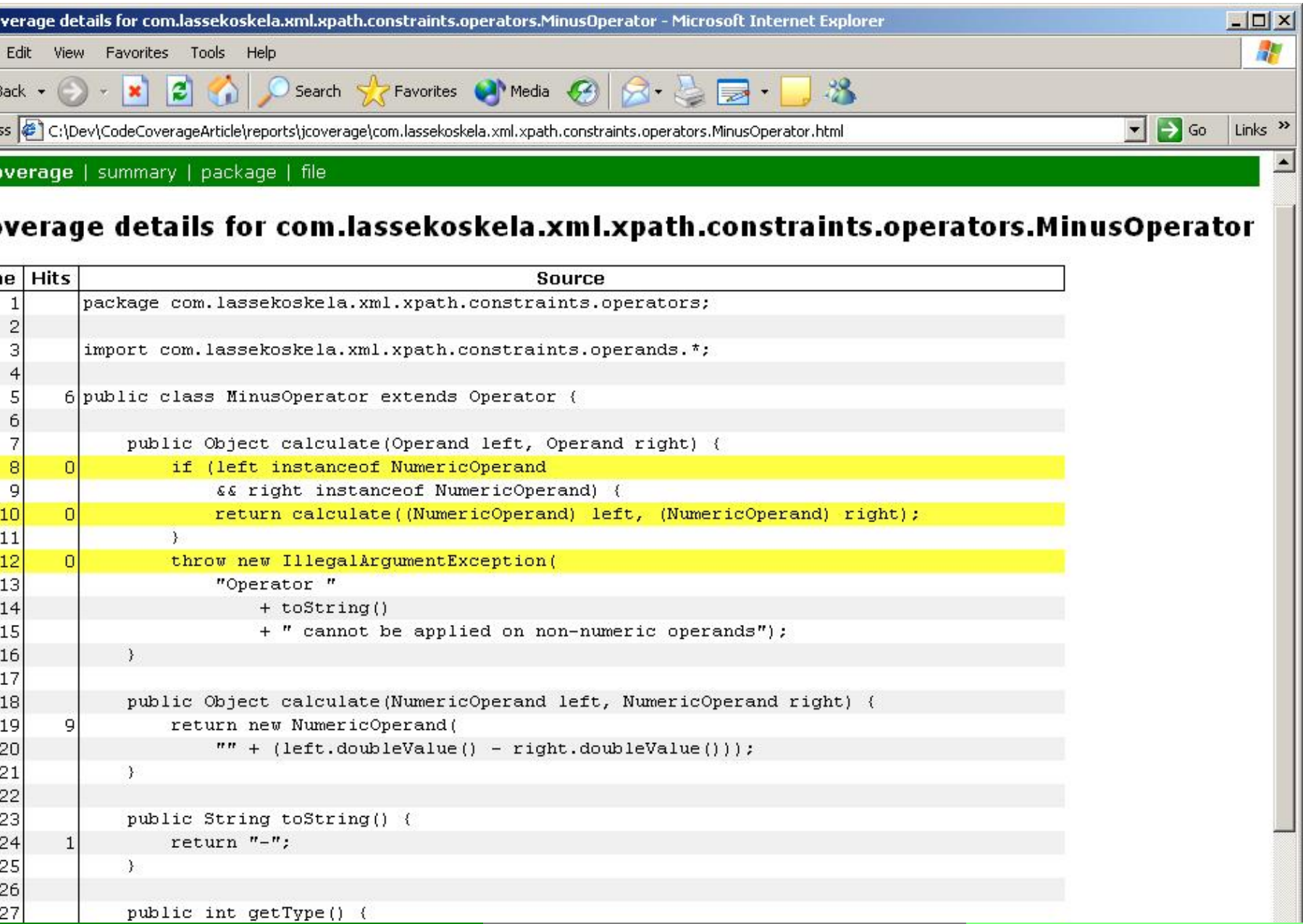
- les tests de couverture sont fait de sorte à passer dans chaque partie du programme

```
if (cas qui arrive très souvent) {
    code qui fonctionne correctement
}
else // cas très rare {
    code avec un gros bug dedans
}
```

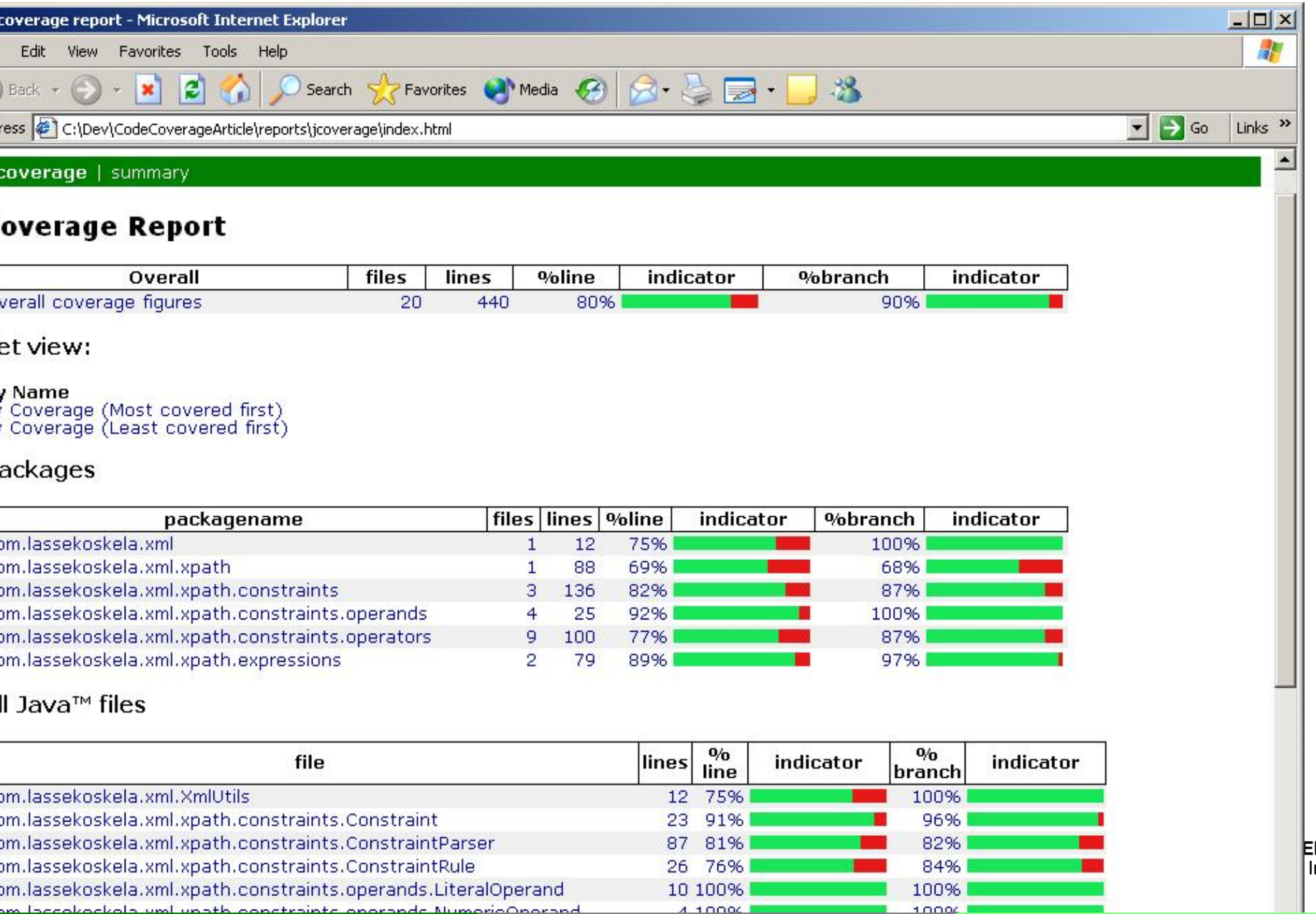
- Principe : concevoir une batterie de tests de sorte à passer sur toutes les lignes du programme.
- Outils :
  - Pour C : purecoverage, checker
  - Pour Java : JCoverage

DÉPARTEMENT Sciences Informatiques

JCoverage


verage details for com.lassekoskela.xml.xpath.constraints.operators.MinusOperator - Microsoft Internet Explorer
Edit View Favorites Tools Help
Search Favorites Media
C:\Dev\CodeCoverageArticle\reports\jcoverage\com.lassekoskela.xml.xpath.constraints.operators.MinusOperator.html
Go Links
verage | summary | package | file
verage details for com.lassekoskela.xml.xpath.constraints.operators.MinusOperator
Hits Source
1 package com.lassekoskela.xml.xpath.constraints.operators;
3 import com.lassekoskela.xml.xpath.constraints.operands.*;
5 6 public class MinusOperator extends Operator {
7 public Object calculate(Operand left, Operand right) {
8 0 if (left instanceof NumericOperand
9 && right instanceof NumericOperand) {
10 0 return calculate((NumericOperand) left, (NumericOperand) right);
11 }
12 0 throw new IllegalArgumentException(
13 "Operator "
14 + toString()
15 + " cannot be applied on non-numeric operands");
16 }
18 public Object calculate(NumericOperand left, NumericOperand right) {
19 9 return new NumericOperand(
20 "" + (left.doubleValue() - right.doubleValue()));
21 }
23 public String toString() {
24 1 return "-";
25 }
27 public int getType() {

JCoverage



EMENT
Informatiques

Tests de performance (bancs d'essai - benchmarks)

idem cas d'utilisation mais sur des cas standards pour comparer les performances avec d'autres produits concurrents
Spécification des tests

- matériels
- système d'exploitation
- contexte (mode graphique/mode console)
- paramétrage
- traces

Tests de non-régression

D'une version à une autre, les tests qui marchent doivent continuer à marcher
Outils :
junit (automatisation des tests)

- XTest (pour comparer les statistiques et les résultats entre deux version automatiquement)
- Log4JUnit

DÉPARTEMENT
Sciences Informatiques

Traces

Informations pertinentes

- date, thread, classes, numéro de ligne
- état mémoire
- paramètres utilisés

Sorties des traces

- Fichiers spéciaux
- affichage écran
- alerte
- comptabilité
- application

Niveau des messages

- debug, info, notification,
- avertissement (warning)
- erreur, erreur critique, alerte, urgence (emergency)

DÉPARTEMENT Sciences Informatiques

Outils de traçage

Utilitaires centralisés de gestion des journaux permettant :

- d'afficher des renseignements
- de spécifier lors de l'execution du programme et non de sa compilation du niveau des messages voulu, leurs sorties et leurs formats.

Il permet d'effectuer aussi bien du 'débuguage' qu'une journalisation des évenements.
Logiciels :

- log4j
- syslog