INGI2132
Assignement 1 Report

Xavier Crochet, Julien De Coster

14 février 2013

1 Introduction

We implemented first the division, the unary plus and the modulo. We wrote likewise the program *primes.java* with the Eratosthenes algorithm. We used jUnit tests to test both the operators and the j– program.

2 Development

The main choice we did was on the data structure of the *primes.java* program. We analyzed two possibilies :

1. A structure with two lists. The first list contains integers from 1 to n. The second list contains the marks. Both lists are equals in size in order to match each element from the first list to a mark from the second one. An element is marked as 0 if it is potentially a prime number or as 1 if it is identified as a multiple of a previous number of the list. We use the term *potentially* because all the elements of the mark list are initialized at 0 except 0 and 1, initialized at 1.
2. A structure with an unique linked list. Here the strategy is to remove every element identified as a multiple of a previous number. The benefit of this method is clear : we don't have to visit elements wich are not relevant because already identied.

We choose the first strategy mainly for performance. Accessing elements and remove them from a linked list is costly. Our mesures showed that the first strategy took more or less ten times less time with wide values (e.g. 50 000).

3 Testing

We tested both the division, plus and modulo with priorities and the j– program.

For the operators we used first basic tests (e.g. "4 / 2" or "4 mod 5") and then we tested the priorities (e.g. "3 + 4 / 5").

We used another strategy for the j– program. We mainly tested the extreme values : "What happen if we put negative numbers ?". We tested for values like 0, 1 (intialized at 1) and 2.