

TD 5

Entiers de Guiseppe Peano

TAD
Semestre 2

1

L^AT_EX ça roxxxxxx trop du poney !

2 Spécification fonctionnelle du TAD Peano

Peano	Entier
zéro	0
succ	+1
add	+
mult	*
inf	<
egal	=

2.1

$$\begin{aligned}
 (p1) \quad 0 + p &= p \\
 (p2) \quad (p + 1) + q &= (p + q) + 1 \\
 (p3) \quad 0 * p &= 0 \\
 (p4) \quad (p + 1) * q &= (p * q) + q \\
 (p5) \quad \neg(0 < 0) \quad (p6) \quad \neg 0 < p + 1 \\
 (p7) \quad \neg(p + 1 < 0) \\
 (p7) \quad p + 1 < q + 1 &\equiv p < q \\
 (p8) \quad p + 1 < q + 1 &\equiv p < q \\
 (p9) \quad 0 &= 0 \\
 (p10) \quad \neg(0 = p + 1) \\
 (p11) \quad \neg(p + 1 = 0) \\
 (p12) \quad p + 1 = q + 1 &\equiv p = q
 \end{aligned}$$

2.2

```

1  add(succ(succ(zero)), succ(succ(succ(zero)))) =
2  succ(add(succ(zero), succ(succ(succ(zero))))) =
3  succ(succ(add(zero, succ(succ(succ(zero))))) =
4  succ(succ(succ(succ(succ(zero))))) =

```

Utile au client

3 Spécification algorithmique du TAD Peano

3.1

```
1  fonction zero()
2      retourne <Peano>;
3
4  fonction succ(entree p <Peano>)
5      retourne <Peano>
6      declenche debordement;
7
8  fonction add(entree p <Peano>, entree q <Peano>)
9      retourne <Peano>
10     declenche debordement;
11
12 fonction mult (entree p <Peano>, entree q <Peano>)
13     retourne <Peano>
14     declenche debordement;
15
16 fonction inf (entree p <Peano>, entree q <Peano>)
17     retourne <Booleen>;
18
19 fonction egal (entree p <Peano>, entree q <Peano>)
20     retourne <Booleen>;
```

3.2

En-tête (cf 3.1) + Propriétés (cf sujet, page 2) + en-tête de l'affectation

3.3

```
1  -- client
2  glossaire
3      p1 <Peano>;
4      p2 <Peano>;
5  debut
6      p1 <- zero;
7      p1 <- p2;
8      si p1 = p2 alors
9          --...
10 fin
```

4 Utilisation du TAD Peano

4.1

```

1  -- en iteratif
2  fonction fact (entree p <Peano>)
3      retourne <Peano>
4      declenche debordement
5  glossaire
6      r <Peano>; --retour
7      i <Peano>; --compteur
8  debut
9      i <- succ(zero);
10     r <- succ(zero);
11     tantque inf(i, p) ou i = p faire
12         r <- mult(r, i);
13         i <- succ(i);
14     fin tantque;
15     retourner r;
16 fin
17
18 -- En recursif :->
19 fonction pred(entree p <Peano>)
20     retourne <Peano>
21 debut
22     si p = zero alors
23         retourner(succ(zero));
24     sinon
25         retourner(mult(fact(pred(p), p)));
26     fin si;
27 fin

```

4.2

```

1
2  fonction infEgal(entree p1 <Peano>,
3                  entree p2 <Peano>)
4  retourne <Booleen>
5  debut
6      retourner inf(p1, succ(p2));
7  fin

```

5 Implémentation du TAD Peano

5.1

```

1  constante N <Entier> = 10 000;
2  type Peano : tableau [1 .. N] de <Caractere>;

```

5.2

```

1  fonction zero ()
2      retourne <Peano>
3  glossaire
4      p <Peano>;
5  debut
6      p[1] <- '!';
7      retourner p ;
8  fin
9
10
11 fonction succ(entree p <Peano>)
12     retourne <Peano>
13     declenche debordement
14 glossaire
15     s <Peano>;
16     i <Entier>;
17 debut
18     i <- 1;
19
20     tantque p[i] < '!' faire
21         s[i] <- '*';
22         i <- i + 1;
23     fin tantque;
24     s[i] <- '*';
25     si i = LG_MAX alors
26         declencher(debordement);
27     fin si;
28
29     s[i+1] = '!';
30     retourner(s)
31 fin

```

5.3 opération add

```

1  -- s = add(p,q)
2  recopier les caracteres '*' de p dans s;
3  recopier les caracteres '*' de q dans s;
4  ajouter le caractere '!' a s;

```

Listing 1 – Algorithme général

```

1  fonction add(entree p <Peano>, entree q <Peano>)
2      retourner <Peano>
3  glossaire
4      i <Entier>;
5      j <Entier>;
6      r <Peano>; -- retour
7
8  debut
9      -- recopier p;
10     i <- 1;
11     tantque p[i] != '!' faire
12         r[i] = '*';
13         i <- i + 1;
14     fin tantque;
15     i <- i - 1;
16
17     --recopier q dans r
18     j <- 1;
19     tantque q[j] != '!' faire
20         si (i+j) > N alors
21             declencher debordement;
22         fin si;
23         r[i+j] <- '*';
24         j <- j+1;
25     fin tantque;
26
27     -- ajouter '!' a s
28     si (i+j) > N alors
29         declencher debordement;
30     fin si; r[i+j] = '!'; retourner r;
31
32 fin

```

Listing 2 – Programme

5.4 opération mult

```

1  se positionner sur le premier caractere '*' de p;
2  tantque il reste une etoile dans p faire
3      recopier les caractere '*' de q dans le produit;
4      passer au caractere '*' suivant de p;
5  fin tantque;

```

Listing 3 – Algorithme général

```

1  fonction mult (entree p <Peano>, entree q <Peano>)
2      retourne <Peano>
3      declenche debordement
4  glossaire
5      i <Entier>;
6      j <Entier>;
7      k <Entier>;
8      r <Peano>;
9
10  debut
11      k <- 1;
12      i <- -1;

```

```
13  tantque p[i] /= '!' faire
14      j <-1;
15      tantque q[j] /= '!' faire
16          si k > N alors
17              declencher debordement;
18          fin si;
19          s[k] <- '*';
20          j <- j + 1;
21          k <- k + j;
22      fin tantque;
23      i <- i + j;
24  fin tantque;
25  si k > N alors
26      declencher debordement;
27  fin si;
28  s[k] <- '!' alors
29      retourner s;
30  fin
```

Listing 4 – Programme