

# TD 4

## Le Type Abstrait de Données Durée

TAD  
Semestre 2

### 1 Spécification algorithmique du Type Abstrait de Données Durée

#### 1.1

```
1  -- Necessite  $h \leq 0$  et  $0 \leq m \leq 59$  et  $0 \leq s \leq 59$ 
2  fonction uneDuree (entree h <Entier>, entree m <Entier>, entree s <Entier>)
3      retourne <Duree>
4      declenche dureeInvalide;
5
6  fonction heures (entree d <Duree>)
7      retourne <Entier>;
8
9  fonction minutes(entree d <Duree>)
10     retourne <Entier>;
11
12 fonction secondes(entree d <Duree>)
13     retourne <Entier>;
14
15 fonction inf(entree d1 <Duree>, entree d2 <Duree>)
16     retourne <Booleen>;
17
18 --exportation de l'egalite
19 fonction "="(entree d1 <Duree> entree d2 <Duree>)
20     retourne <Booleen>;
21
22 procedure ajouterUneSeconde(maj d <Duree>);
```

#### 1.2

```
1  --En-tetes + proprietes (sujet)
2
3  -- Necessite  $h \leq 0$  et  $0 \leq m \leq 59$  et  $0 \leq s \leq 59$ 
4  fonction uneDuree (entree h <Entier>, entree m <Entier>, entree s <Entier>)
5      retourne <Duree>
6      declenche dureeInvalide;
7
8  fonction heures (entree d <Duree>)
9      retourne <Entier>;
10
11 fonction minutes(entree d <Duree>)
12     retourne <Entier>;
13
14 fonction secondes(entree d <Duree>)
```

```

15     retourne <Entier>;
16
17 fonction inf(entree d1 <Duree>, entree d2 <Duree>)
18     retourne <Booleen>;
19
20 procedure ajouterUneSeconde(maj d <Duree>);
21
22 --exportation de l'egalite
23 fonction "="(entree d1 <Duree> );
24     retourne <Booleen>;
25
26 --exportation de l'affectation
27 procedure "<-"(entree d1 <Duree> entree d2 <Duree>);

```

## 2 Implémentation du Type Abstrait de Données Durée

### 2.1

```

1 type Duree: enregistrement
2     h <Entier>,
3     m <Entier>,
4     s <Entier>;

```

### 2.2

```

1 fonction estValide (entree d <Duree>) retourne <Booleen>
2 debut
3     retourner d.h < 0 ou d.m < 0 ou d.s < 0 ou d.m > 59 ou d.s > 59 ;
4 fin
5
6 -- Necessite  $h \leq 0$  et  $0 \leq m \leq 59$  et  $0 \leq s \leq 59$ 
7 fonction uneDuree (entree h <Entier>, entree m <Entier>, entree s <Entier>)
8     retourne <Duree>
9     declenche dureeInvalide
10 glossaire
11     d <Duree>;
12 debut
13     si non estValide(d) alors
14         declencher dureeInvalide;
15     fin si;
16
17     d.h <- h;
18     d.m <- m;
19     d.s <- s;
20
21     retourner d;
22 fin
23
24 fonction heures (entree d <Duree>)
25     retourne <Entier>
26 debut
27     retourner d.h ;
28 fin
29
30 fonction minutes(entree d <Duree>)

```

```

31     retourne <Entier>
32 debut
33     retourner d.m ;
34 fin
35
36 fonction secondes(entree d <Duree>)
37     retourne <Entier>
38 debut
39     retourner d.s ;
40 fin
41
42 fonction inf(entree d1 <Duree>, entree d2 <Duree>)
43     retourne <Booleen>
44 debut
45     retourner d1.h * 3600 + d1.m * 60 + d1.s < d2.h * 3600 + d2.m * 60 + d2.s;
46     -- ou sinon
47         --retourner ((d1.h<d2.h) ou (d1.h=d2.h et d1.m < d2.m)
48         -- ou (d1.h = d2.h et d1.m = d2.m et d1.s < d2.s));
49 fin
50
51 -- exportation de l'egalite
52 -- deja realise par le compilo => comparaison de deux enregistrement!
53 fonction "="(entree d1 <Duree>, entree d2 <Duree>)
54     retourne <Booleen>;
55
56 --exportation de l'affectation
57 -- deja realise par le compilo => affectation de deux enregistrement!
58 procedure "<-"(maj d1 <Duree> entree d2 <Duree>)
59 debut
60 fin
61
62 procedure ajouterUneSeconde (maj d <Duree>)
63 debut
64     si d.s < 59 alors
65         d.s <- d.s + 1;
66     sinon
67         si d.m < 59 alors
68             d.m <- d.m + 1;
69             d.s <- 0;
70         sinon
71             d.h <- d.h + 1;
72             d.m <- 0;
73             d.s <- 0;
74         fin si;
75     fin si;
76 fin

```

## 2.3

Implémentation du TAD Durée :

- type Durée (2-1)
- CORps des sous programmes (2-2)
- Le corps du sous programme estValide

## 2.4

```

1  type Duree : enregistrement
2    s <Entier>;

```

## 2.5

```

1  fonction estValide (entree h <Entier>, entree m <Entier>, entree s <Entier>)
    retourne <Booleen>
2  debut
3    retourner h < 0 ou m < 0 ou s < 0 ou m > 59 ou s > 59 ;
4  fin
5
6  fonction uneDuree(entree h <Entier>, entree m <Entier>, entree s <Entier>)
7    retourne Duree
8    declenche dureeInvalide
9  glossaire
10   d <Duree>;
11  debut
12   si non dureeValide(h, m, s) alors
13     declencher dureeInvalide;
14   fin si;
15   d.s = h*3600 + m * 60 + s
16   retourner d;
17  fin
18
19  fonction heures (entree d <Duree>)
20    retourne <Entier>
21  debut
22    retourner (d.s div 3600);
23  fin
24
25  fonction minutes(entree d <Duree>)
26    retourne <Entier>
27  debut
28    retourner (d.s - heures(d)*3600 div 60) ;
29  fin
30
31  fonction secondes(entree d <Duree>)
32    retourne <Entier>
33  debut
34    retourner d.s mod 60;
35  fin
36
37  fonction inf(entree d1 <Duree>, entree d2 <Duree>)
38    retourne <Booleen>
39  debut
40    retourner d1.s < d2.s;
41  fin
42
43  -- exportation de l'egalite
44  -- deja realise par le compilo => comparaison de deux enregistrement!
45  fonction "="(entree d1 <Duree>, entree d2 <Duree>)
46    retourne <Booleen>;
47
48  --exportation de l'affectation
49  -- deja realise par le compilo => affectation de deux enregistrement!
50  procedure "<-"(maj d1 <Duree> entree d2 <Duree>);

```

```
51
52 procedure ajouterUneSeconde (maj d <Duree>)
53 debut
54     d.s <- d.s + 1;
55 fin
```

## 2.6

Spécification cf question 1.2(valide pour l'implémentation 2.3)

Implémentation -> cd 2.4 et 2.5