MARGUERITE Alain
RINCE Romain

Université de Nantes
2 rue de la Houssinière, BP92208, F-44322 Nantes cedex 03, FRANCE

Encadrant : Christophe JERMANN

1 Tests de sous classes concrètede collection et map

Cette classe hérite de la classe Abstract LIST[G] definie en eiffel. Nous avons testés 3 de ces classes :

- linkedlist
- array
- arrayed_list

Initialisations de ces collection et utilisations :

```
liste:LINKED_LIST[PERSONNE]
liste1:ARRAY[PERSONNE]
liste3:ARRAYED_LIST[PERSONNE]
create liste1.make(0,10);
create liste.make();
```

Les insertions sont faites simplement à l'aide de la methode out de list :

```
liste.put_front (p);
liste.put(m);
liste.put(n);
liste.put(s);
```

Elle permet de stocker les noms des joueurs et d'autres composants du jeu. Elle presente des fonctionalités qui permettent un accès aux elements via la méthode : item :G :renvoie l'element courant Le parcours de notre LINKED_List peut se faire : start :qui deplace le cursor à la premiere position et renvoie un boolean exhausted :renvoie un boolean pour dire que la collection a ete totalement parcourue

```
from  liste.start
```

```
until liste.exhausted
loop
print("NOM:" +liste.item.nom+ "%N");
print("PRENOM" +liste.item.prenom+"%N");
print("AGE:" +liste.item.age.out+"%N");
liste.forth
end
```

Dans un ARRAY[G],on donne l'indice de l'item qu'on désire récuperer. Pour un parcours on utilise incrementation de cet indice

```
from i:=0
until i<10
loop
print("NOM:" +liste1.item (i).nom+ "%N");
print("PRENOM" +liste1.item (i).prenom+"%N");
print("AGE:" +liste1.item (i).age.out+"%N");
i:=i+1;
    end
```

2 Manipulation de fichiers et découpage de chaînes de caractères

2.1 Fichiers

Tests sur des methodes de la librairie standard d'Eiffel pemettant la lecture et l'écriture en eiffel. Les objets de classe PLAIN_TEXT_FILE semblent adaptés à cet usage.

```
create ifile.make_open_read ("test_input.txt")
```

Cette methode permet d'ouvrir le fichier source avec naturellement les droits de lectures indispensables. Il est nécessaire de placer le fichier source dans le répertoire courant. De même le fichier de sortie sera place dans ce repertoire.

D'autres méthodes sont particulièrement utiles comme :

```
ifile.exhausted
```

Celle-ci renvoit un booléen indiquant si l'on se trouve en fin de fichier ou non.

Lorsque l'objet ifile a lu le caractère, un de ses attributs (last_character)a pris la valeur de ce caractère. Pour écrire dans le fichier il suffit donc d'appeller la méthode put de ofile sur cet attribut :

```
ofile.put (input_file.last_character)
```

L'écriture dans le fichier est intuitive, il suffit de chercher la methode adéquate. Par exemple écrire une string dans le fichier :

```
ofile.putstring ("Fin du fichier")
```

2.2 Strings

Le découpage de caractère est facile par les méthodes de l'objet string mises à disposition. En effet, on trouve différentes méthodes similaires à :

```
substring (start_index, end_index: INTEGER_32)
```

On peut ainsi créer directement plusieurs chaînes à partir d'un découpage donné par des index :

```
!!ma_chaine2.make_from_string (ma_chaine.substring (1,3))
```