

### Exemple : une copie de tableaux

```
/** CDI cours B3, ED 14, Ex06 Hoare et tableaux */
public class CopieDeTableaux {

    public static void main(/**@ non_null */String[] args){
        int[] t = {1,9,11,15,56,78,89};
        /**@ assert (\exists int x: x>=0 & x<t.length & t[x]==89);
        int[] t1 = new int[t.length];
        /**@ assert t != null & t1 != null;
        /**@ assert t.length == t1.length;
        int i = 0;
        int n = t.length-1;
        /**@ loop_invariant (\forall int j: j>=0 & j < i ==> t[j] == t1[j]);
        while( i <= n){
            t1[i] = t[i];
            i++;
        }
        /**@ assert (\forall int j: j>=0 & j <= n ==> t[j] == t1[j]);
    }
}
```

ESC\_Java

1

### La pile (encore), les pré-conditions

```
public class File{
    private int[] zone;
    private int ptr;

    /**@ requires taille > 0;

    public File(int taille){
        zone = new int[ taille];
    }

    /**@ requires ptr < zone.length; // !estPleine();

    public void empiler(int elt){
        zone[ptr] = elt;ptr++;
    }

    /**@ requires ptr > 0; // !estVide();

    public int depiler(){
        ptr--;return zone[ptr];
    }
}
```

ESC\_Java

2

### La pile, /\*@ spec\_public \*/

```
public class File{
    private /*@ spec_public */ int[] zone;
    private /*@ spec_public */ int ptr;

    /**@ requires taille > 0;

    public File(int taille){
        zone = new int[ taille];
    }

    /**@ requires ptr < zone.length;

    public void empiler(int elt){
        zone[ptr] = elt;ptr++;
    }

    /**@ requires ptr > 0;

    public int depiler(){
        ptr--;return zone[ptr];
    }
}
```

ESC\_Java

3

### La pile, les post conditions

```
public class File{
    private /*@ spec_public */ int[] zone;
    private /*@ spec_public */ int ptr;

    /**@ requires taille > 0;
    /**@ ensures zone != null & ptr == 0 && zone.length == taille;
    public File(int taille){
        zone = new int[ taille];
    }

    /**@ requires ptr < zone.length;
    /**@ modifies ptr, zone[ptr-1];
    /**@ ensures \old(ptr) + 1 == ptr && zone[\old(ptr)] == elt;
    public void empiler(int elt){
        zone[ptr] = elt;ptr++;
    }

    /**@ requires ptr > 0;
    /**@ modifies ptr;
    /**@ ensures \old(ptr)-1==ptr && \result ==zone[ptr];
    public int depiler(){
        ptr--;return zone[ptr];
    }
}
```

ESC\_Java

4

### La pile, invariant

```
public class File{
    private /*@ spec_public */ int[] zone;
    private /*@ spec_public */ int ptr;
    /**@ invariant ptr >= 0 && ptr <= zone.length;
    /**@ invariant zone != null;

    /**@ requires taille > 0;
    /**@ ensures ptr == 0 && zone.length == taille;
    public File(int taille){
        zone = new int[ taille];
    }

    /**@ requires ptr < zone.length;
    /**@ modifies ptr, zone[ptr-1];
    /**@ ensures \old(ptr) + 1 == ptr && zone[\old(ptr)] == elt;
    public void empiler(int elt){
        zone[ptr] = elt;ptr++;
    }

    /**@ requires ptr > 0;
    /**@ modifies ptr;
    /**@ ensures \old(ptr)-1==ptr && \result ==zone[ptr];
    public int depiler(){
        ptr--;return zone[ptr];
    }
}
```

ESC\_Java

5