2019-07-18

ES_1

{a,b,c}, {c,d,e} ⟶ {a,b,c}
1 2 = m.length {c,d,e}

boolean method (int[][] m)

```
/* @ requires m!= null
   @ ensures (result == true) <==> (\forall int i; i>=0 && i < m.length;
   @                                  (\forall int j; j>=0 && j< m[i].length-1; m[i][j] < m[i][j+1]))
   @                                  &&
   @                                (\forall int c; c >=0 && < m[c].length;
   @                                  (\forall int r; r>=0 && r < m.length-1 ; m[r][c] < m[r+1][c]));
   @
 */
```

boolean method (int[] a)

```
/* @ requires a!= null && a.length > 0;
   @ ensures (\result == true) <==> (\forall int i; i > 0 && i < a.length-1; a[i+1] == 2*a[i]) &&
   @                                ( a[a.length-2] == (\sum int i; i>=0 && i < a.length-1; a[i]));
   @
 */
```

boolean method (String s1, String s2) throws NullArgumentException

```
/* @ requires true
   @ ensures (result == true) <==> ( \forall int i; i>=0 && i < s2.length;
   @                                   (\exists int j; j>=0 && j< s1.length; s1[j] == s2[i]));
   @ Signals (NullArgumentException nae) s1 == null || s2==null;
 */
```

int method (ArrayList<> a)

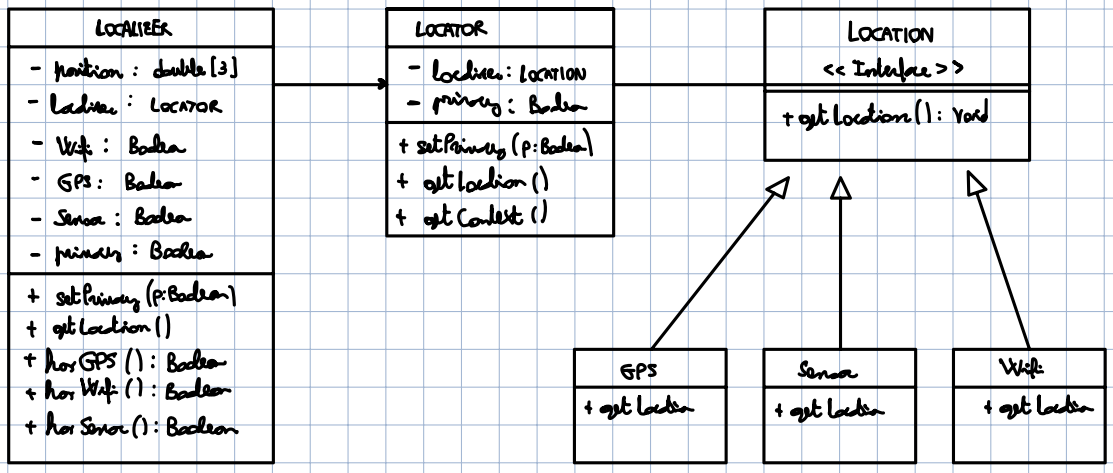
```
/* @ requires a!= null && a.size()>0;
   @ ensures (\result == (\max( \forall int i; i>=0 && i< a.size();
   @                              (\num_of int k; k>=0 && k<a.size && k!=i; a[i]==a[k])));
   @
 */
```

ES_2

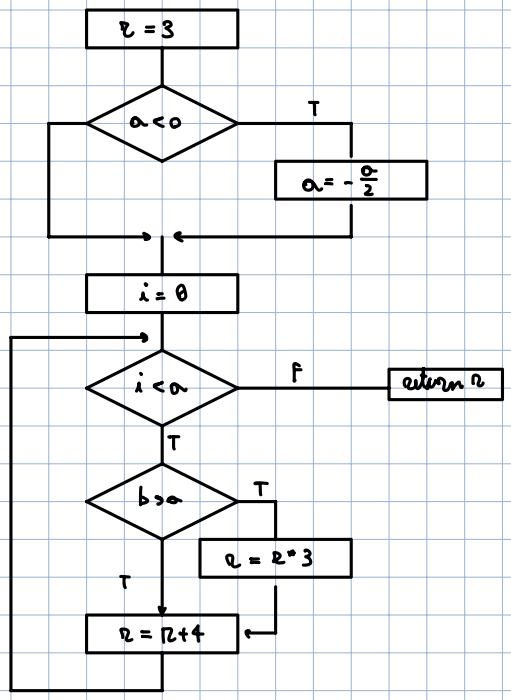
```
public class Wallet implements Runnable {
    private List<Stock> stocks;

    public Wallet(List<Stock> stocks) { this.stocks = stocks; }
    public void update(Stock s, Double n) {
        if(stocks.contains(s)){
            stocks.remove(s);
            s.setValue(n);
            stocks.add(s);
            notifyAll();
        }
    }
    public synchronized double avg(){
        return stocks.stream().mapToDouble(s -> s.getRoi()).average().orElse(0);}

    @Override
    public void run(){
        double avg;
        while((avg = getAvg()) <= 0){
            try { wait();
            catch (Exception e) {}
        }
        System(avg);
        try { wait();
        catch (Exception e) {}
    }
```

```
public class Stock implements Runnable {
    private String name;
    private Wallet wallet;
    private Double value;
    private double roi;
    public Stock(Wallet wallet, String name, Double value){
        ...
    }

    public void setVal(double newVal){
        roi = newVal - value;
        value = newVal;
    }

    @Override
    public void run(){
        wallet.update(this, random);
        try { Thread.sleep(random);}
        catch(...) {}
    }
```

ES-3


LOCALIZER
- position : double[3]
- locator : LOCATOR
- Wifi : Boolean
- GPS : Boolean
- Sensor : Boolean
- primary : Boolean
+ setPrimary(p:Boolean)
+ getLocation()
+ hasGPS() : Boolean
+ hasWifi() : Boolean
+ hasSensor() : Boolean
LOCATOR
- location : LOCATION
- primary : Boolean
+ setPrimary(p:Boolean)
+ getLocation()
+ getContext()
LOCATION
<< Interface >>
+ getLocation() : void
GPS
+ getLocation
Sensor
+ getLocation
Wifi
+ getLocation

ES-5


r = 3
a < 0
T
a = -a/2
i = 0
i < a
F
return r
T
b > a
T
r = r * 3
T
r = r + 1