

2022-08-29

ES-1

```
int /* @pure */ method (String s)
```

```
/* @requires s != null ;
   @ensures (result == (num. of int i; i >= 0 && i < s.length();
   @           !(exists int j; j >= 0 && j < s.length() && j != i;
   @           s.charAt(i).equals(s.charAt(j)))));
   */
```

ES-2

Date: Abstract \rightarrow date Distance (Date)

\downarrow

DayDate: \rightarrow date Distance (Date)

\downarrow

date Distance (DayDate)

DayMonthDate \rightarrow date Distance (Date) \rightarrow date Distance (DMD) D)

\downarrow

date Distance (DMD) \rightarrow super. Distance

DayMonthYearDate \rightarrow date Distance (Date)

\downarrow

date Distance (DMYD)

d1: DD \rightarrow DD 22

d2: DD \rightarrow DMD 26, 3

d3: DMD \rightarrow DMD 14, 9

d4: D \rightarrow DMD 12, 5

d5: DMYD \rightarrow DMYD 18, 4, 2021

~~d6: D \rightarrow D non è possibile ritornare una classe astratta~~

d1. d2 \Rightarrow DD. DMD \Rightarrow 22 - 26 = -4

d1. d3 \Rightarrow DD. DMD \Rightarrow 22 - 14 = 8

d2. d3 \Rightarrow DD. DMD \Rightarrow 26 - 14 = 12

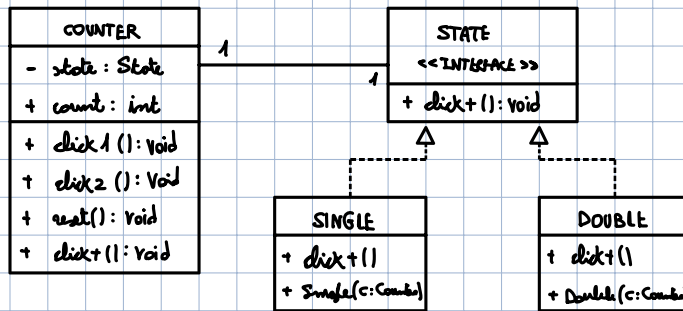
d3. d2 \Rightarrow DMD. DD \Rightarrow DD. DD \Rightarrow 14 - 26 = -12

d3. d5 \Rightarrow DMD. DMYD \Rightarrow -4, 5

d4. d5 \Rightarrow D. DMYD \Rightarrow DMD. DMYD \Rightarrow -6, 1

d5. d3 \Rightarrow DMYD. DMD \Rightarrow 4, -5

ES-3



```

public void click1() {
    this.state = new Single();
}
  
```

```

public void click2() {
    this.state = new Double();
}
  
```

```

public void click() {
    this.state.click();
}
  
```

```

SINGLE :
    void click() {
        this.c.count += 1;
    }
  
```

```

DOUBLE :
    void click() {
        this.c.count += 2;
    }
  
```

ES-4

```

public class ParkingLot {
    private int cars = 5;

    public synchronized boolean getParkingSpot() {
        if (cars < 5) { cars--; return false; }
        return true;
    }

    public synchronized boolean leaveParkingSpot() {
        cars++;
    }
}
  
```

```

public class Car {
    private ParkingLot park;

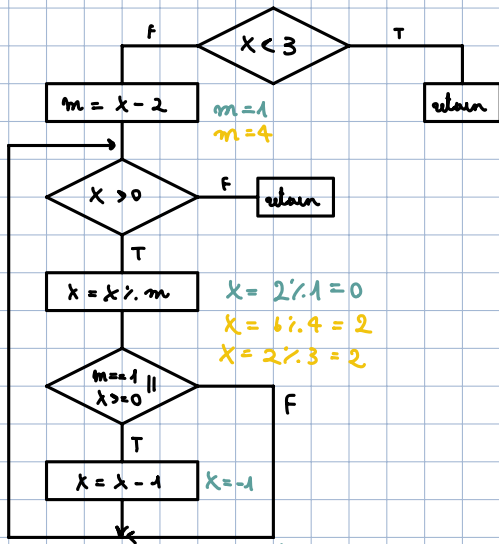
    public Car(ParkingLot park) {
        this.park = park;
    }

    @Override
    public void run() {
        while (!getParkingSpot()) {
            try { Thread.sleep(30000); }
            catch (Exception e) {}
        }

        try { Thread.sleep(2000); }
        catch (Exception e) {}

        park.leaveParkingSpot();
    }
}
  
```

ES-5



• Intension: $\langle 2 \rangle$

$\langle 3 \rangle$

→ esigue anche il
while una sola
volta

• Branch: $\langle 2 \rangle$

$\langle 3 \rangle$

↳ branch non equilibri: $x \geq 0$ sempre