



# JAVA EXCEPTION: UNA VISIONE OPERAZIONALE

1



```

class C {
    public void via() {
        primo();
        System.out.println("siamo in via");
    }

    public void primo() {
        secondo();
        System.out.println("siamo in primo");
    }

    public void secondo () {
        throw new RuntimeException();
        // System.out.println("siamo in secondo");
    }
}

```

2

## Abstract Stack Machine



Workspace

Stack

Heap

(new C()).via();

3

## Abstract Stack Machine



Workspace

Stack

Heap

(new C()).via();

4

## Abstract Stack Machine



Workspace

Stack

Heap

() .via();



**Allocata una istanza della classe C sullo heap.**

5

## Abstract Stack Machine



Workspace

Stack

Heap

() .via() (1);



```

public void via() {
    primo();
    System.out.println("siamo in via");
}

```

6

### Abstract Stack Machine

- Viene copiato nel workspace il corpo del metodo **via()**
- Viene salvata sullo stack la **continuazione**, cioè cosa eseguire dopo aver completato "via" (in questo caso, niente)
- Viene salvato sullo stack anche il valore corrente di **this**.

10 7

### Abstract Stack Machine

```
public void primo() {
    secondo();
    System.out.println("siamo in primo");
}
```

10 8

### Abstract Stack Machine

9

### Abstract Stack Machine

```
public void secondo () {
    throw new RuntimeException();
    System.out.println("siamo in secondo");
}
```

10

### Abstract Stack Machine

11

### Abstract Stack Machine

12

### Abstract Stack Machine

Workspace

```
throw ();
S.out.println("Siamo in secondo");
```

Stack

```
_;
this
_;
S.out.println("Siamo in via");
this
_;
S.out.println("Siamo in primo");
this
```

Heap

```
C
RuntimeEx
```

13

### Abstract Stack Machine

Workspace

Stack

```
_;
this
_;
S.out.println("Siamo in via");
this
_;
S.out.println("Siamo in primo");
this
```

Heap

```
C
RuntimeEx
```

Eliminare il codice corrente dal workspace

Ricerca sullo stack del try-catch piu' recente

Altrimenti , abort con errore

14

### Abstract Stack Machine

Workspace

Stack

```
_;
this
_;
S.out.println("Siamo in via");
this
_;
S.out.println("Siamo in primo");
this
```

Heap

```
C
RuntimeEx
```

Eliminare il codice corrente dal workspace

Ricerca sullo stack del try-catch piu' recente

Altrimenti , abort con errore

TRY-CATCH?

NO!

15

### Abstract Stack Machine

Workspace

Stack

```
_;
this
_;
S.out.println("Siamo in via");
this
_;
S.out.println("Siamo in primo");
this
```

Heap

```
C
RuntimeEx
```

Eliminare il codice corrente dal workspace

Ricerca sullo stack del try-catch piu' recente

Altrimenti , abort con errore

TRY-CATCH?

NO!

16

### Abstract Stack Machine

Workspace

Stack

```
_;
this
_;
S.out.println("Siamo in via");
this
_;
S.out.println("Siamo in primo");
this
```

Heap

```
C
RuntimeEx
```

Eliminare il codice corrente dal workspace

Ricerca sullo stack del try-catch piu' recente

Altrimenti , abort con errore

TRY-CATCH?

NO!

17

### Abstract Stack Machine

Workspace

Stack

Heap

```
C
RuntimeEx
```

Programma termina  
Eccezione non catturata

```
$ java C
Exception in thread "main"
java.lang.RuntimeException
at C.secondo(C.java:13)
at C.primo(C.java:8)
at C.via(C.java:3)
at C.main(C.java:18)
```

18

```

class C {
    public void via() {
        primo();
        System.out.println("siamo in via");
    }

    public void primo() {
        try {secondo();}
        catch (Exception e) { System.out.println("catturata");}
        System.out.println("siamo in primo");
    }

    public void secondo () {
        throw new RuntimeException();
        // System.out.println("siamo in secondo");
    }
}
    
```

Cosa succede (new C()).via();?

### Abstract Stack Machine

L'esecuzione del try-catch inserisce sullo stack il codice di gestione e lascia sul workspace il codice monitorato

### Abstract Stack Machine

### Abstract Stack Machine

- Viene copiato nel workspace il corpo della clausola **catch()**, seguito dal resto del corpo del metodo

### Abstract Stack Machine

```

$ java C
catturata
siamo in primo
siamo in via
    
```