
PROGRAMMAZIONE 2

6b. Le eccezioni, operazionalmente

Un esempio

```
public class C {  
    public void via(){  
        primo();  
        System.out.println("Sei al via");  
    }  
  
    public void primo(){  
        secondo();  
        System.out.println("Sei al primo");  
    }  
  
    public void secondo(){  
        throw new Exception("Prova");  
        System.out.println("Sei al secondo");  
    }  
}
```

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

Abstract Stack Machine

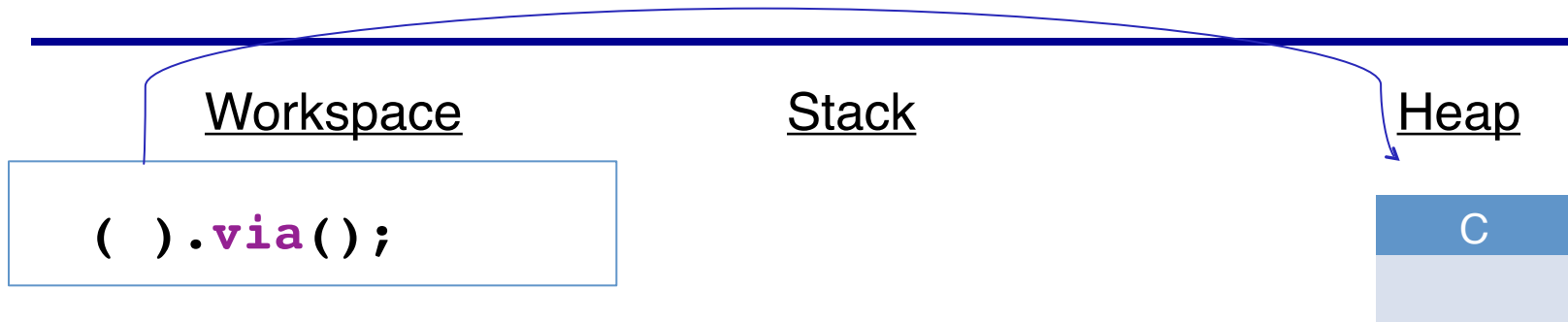
Workspace

```
(new C()).via();
```

Stack

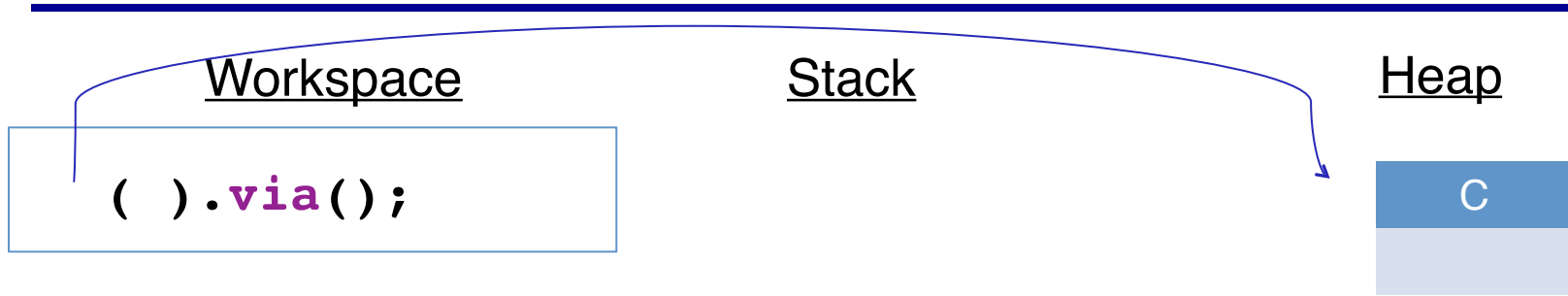
Heap

Abstract Stack Machine



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

Abstract Stack Machine

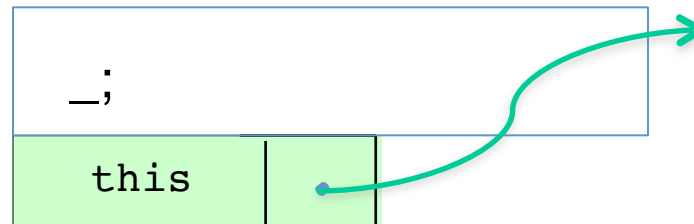


Abstract Stack Machine

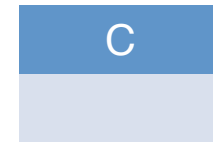
Workspace

```
primo();  
S.o.println("Sei al via");
```

Stack



Heap



Viene salvato sullo stack la continuazione (cosa eseguire) dopo aver invocato **via()**.

Viene salvato sullo stack anche il valore corrente di **this**

Abstract Stack Machine

Workspace

Stack

Heap

```
primo();  
S.o.println("Sei al via");
```

–;

this

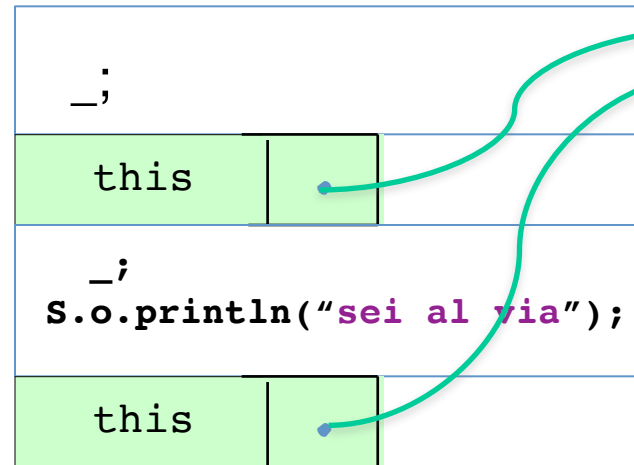
C

Abstract Stack Machine

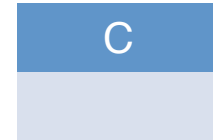
Workspace

```
secondo();  
S.o.println("Sei al primo");
```

Stack



Heap

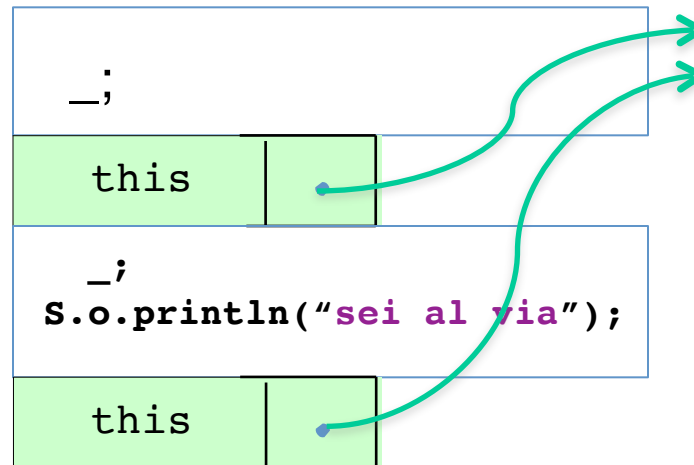


Abstract Stack Machine

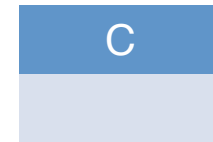
Workspace

```
secondo();  
S.o.println("Sei al primo");
```

Stack



Heap

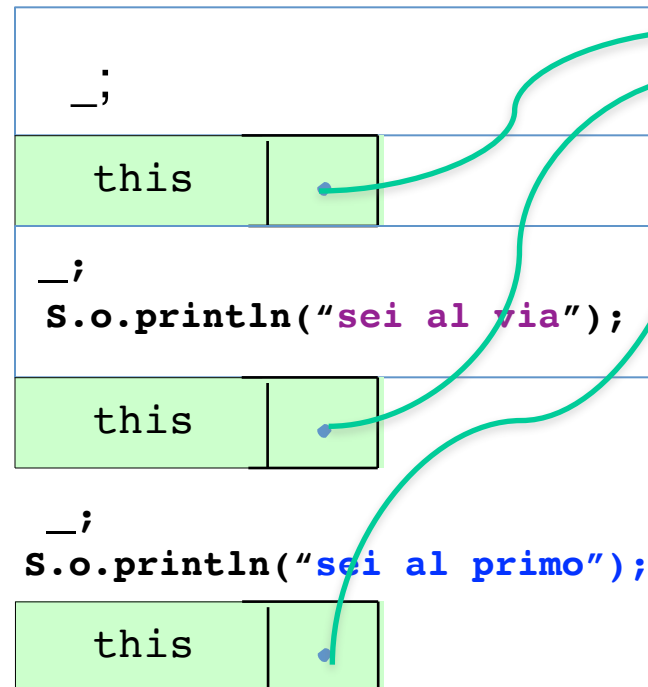


Abstract Stack Machine

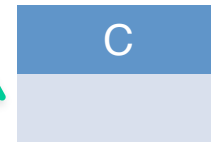
Workspace

```
throw new Exception();
S.o.println("sei al secondo");
```

Stack



Heap

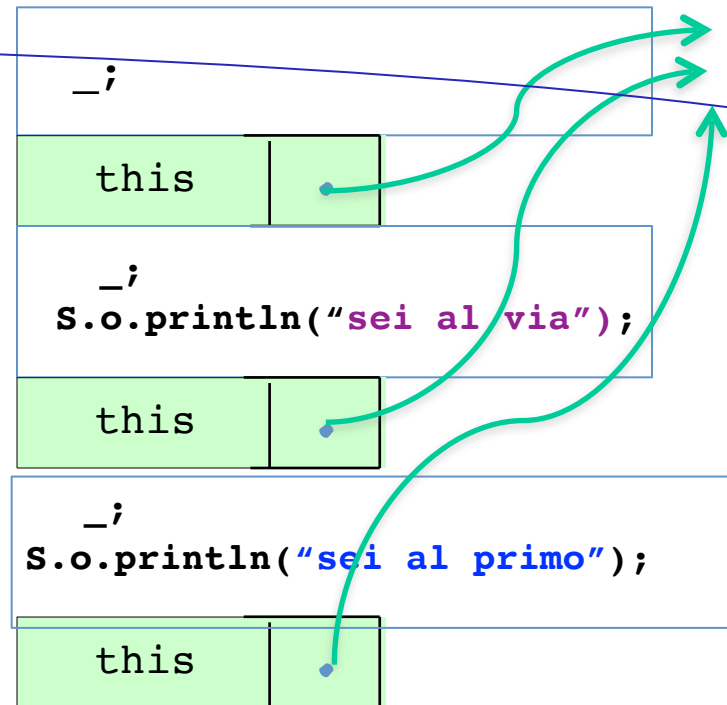


Abstract Stack Machine

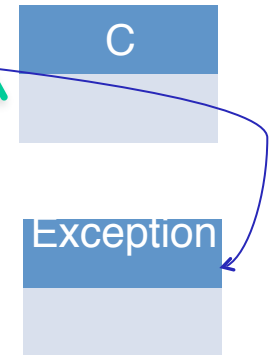
Workspace

```
throw;  
S.o.println("sei al secondo");
```

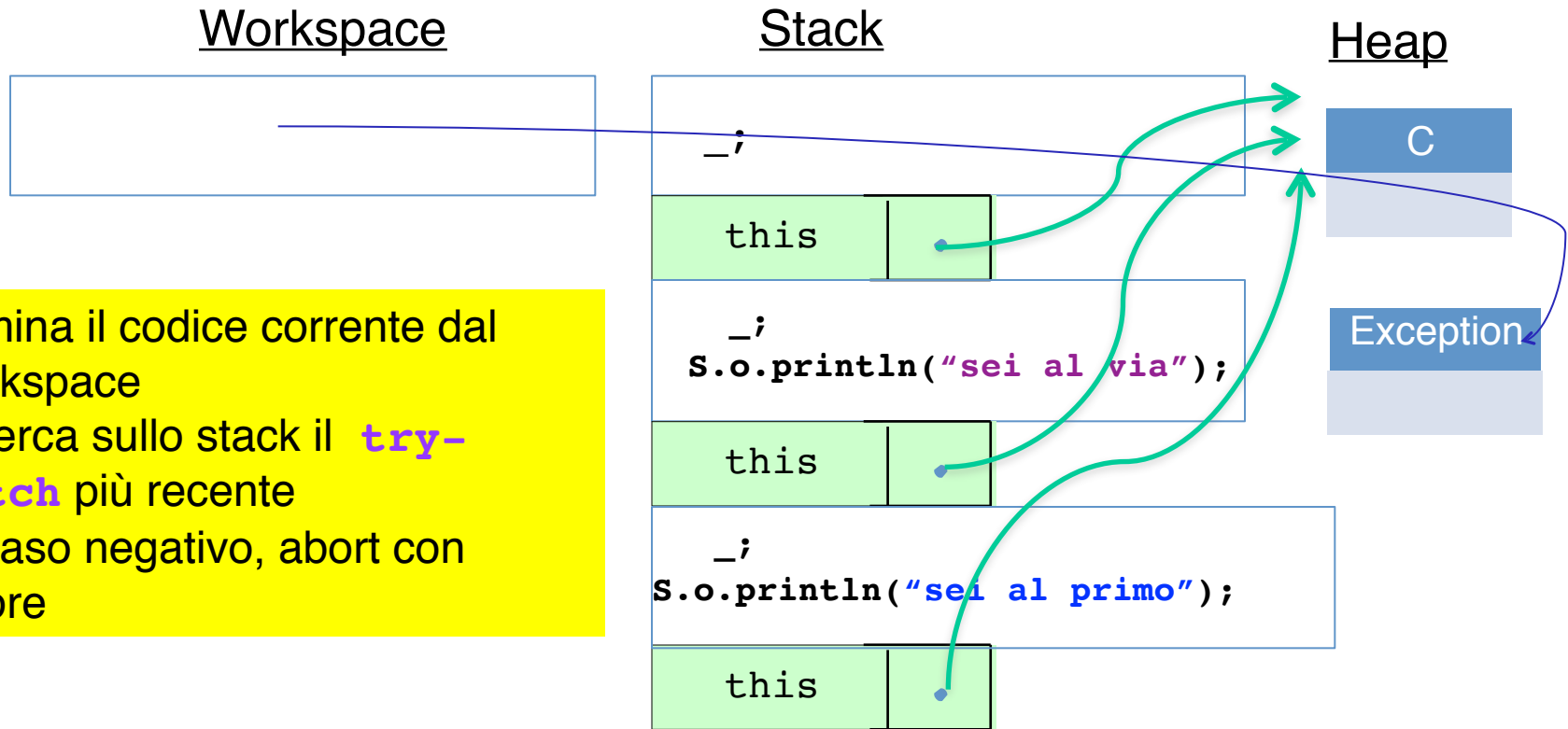
Stack



Heap

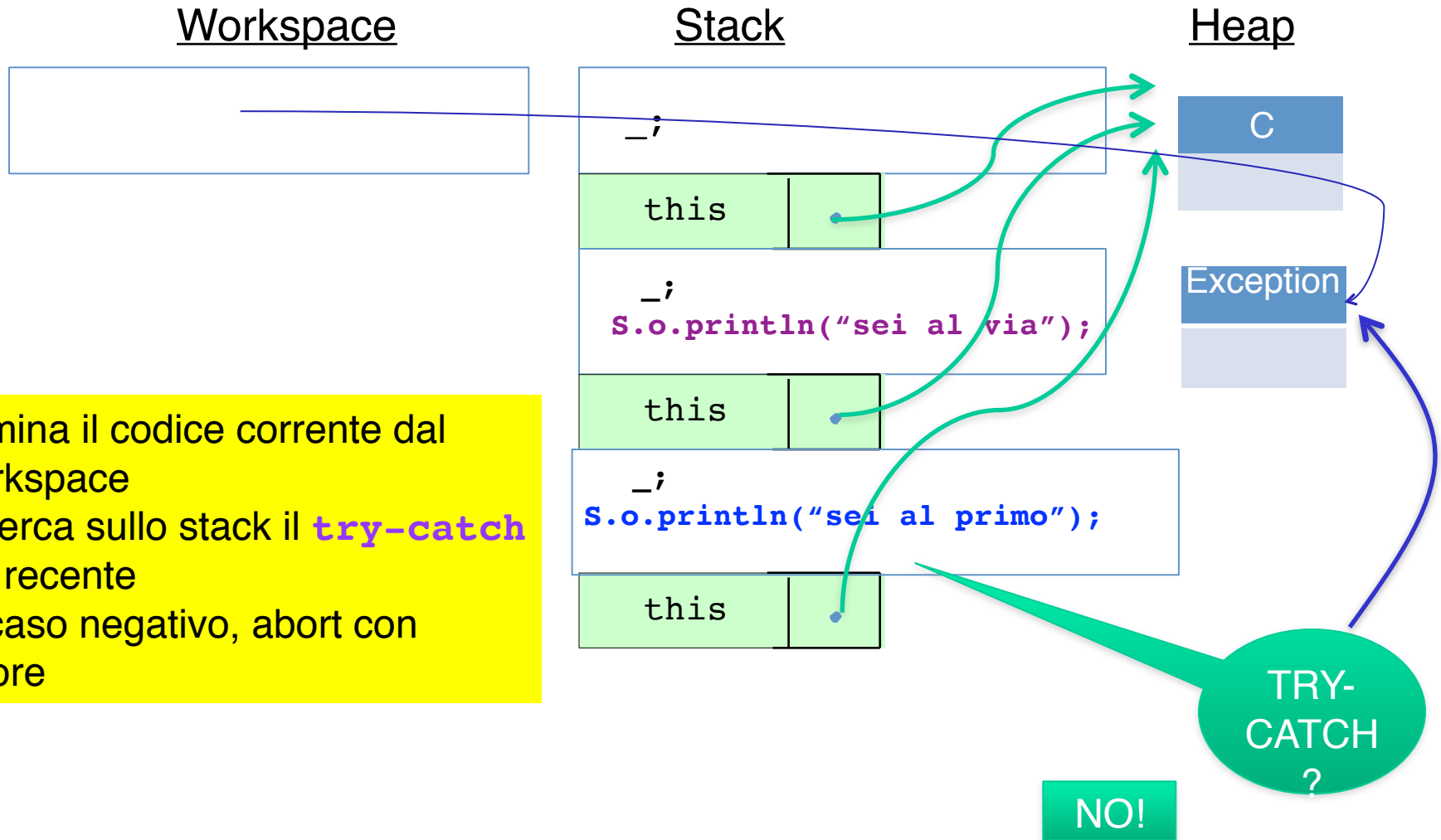


Abstract Stack Machine



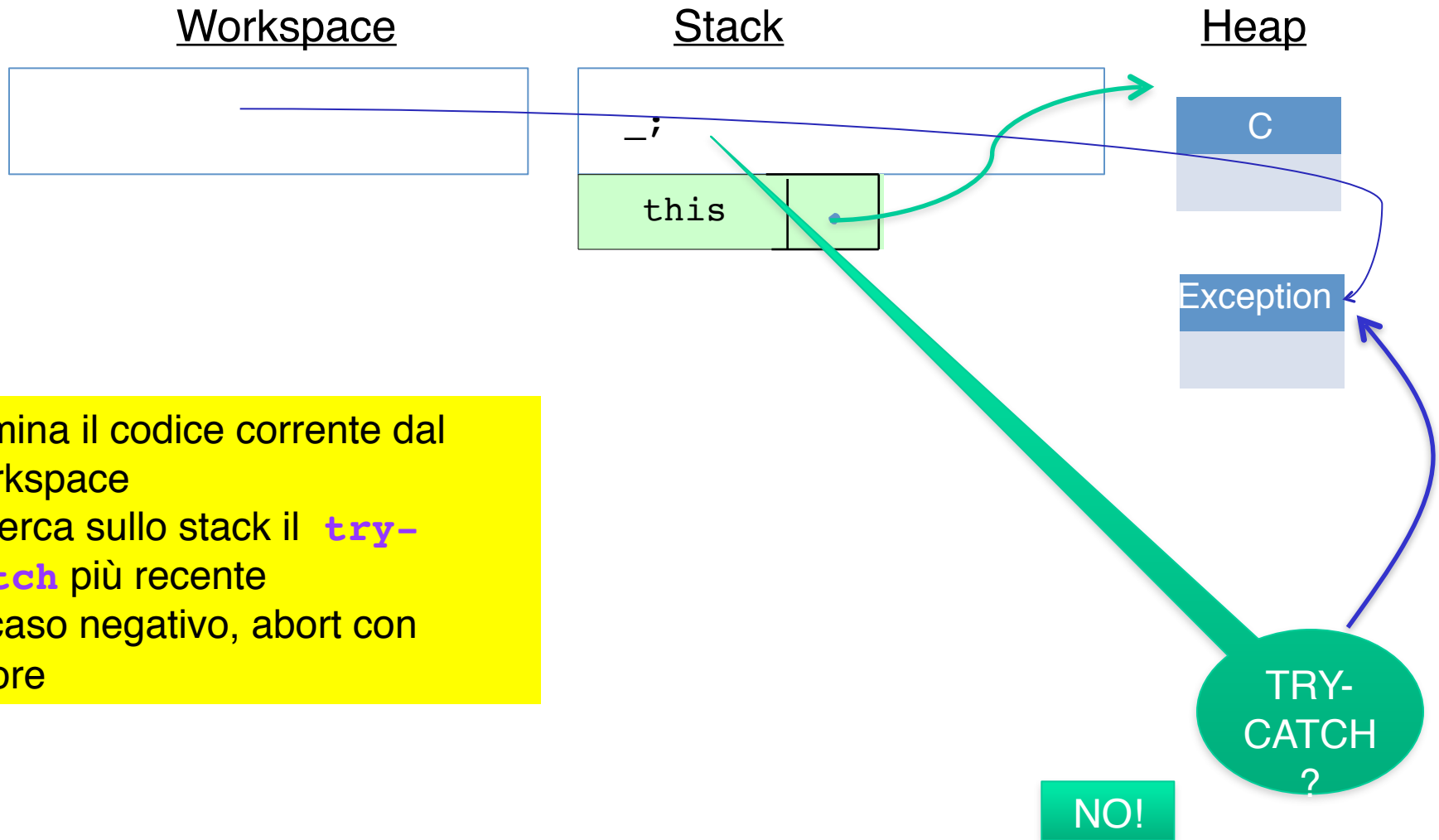
Elimina il codice corrente dal workspace
 Ricerca sullo stack il **try-catch** più recente
 In caso negativo, abort con errore

Abstract Stack Machine

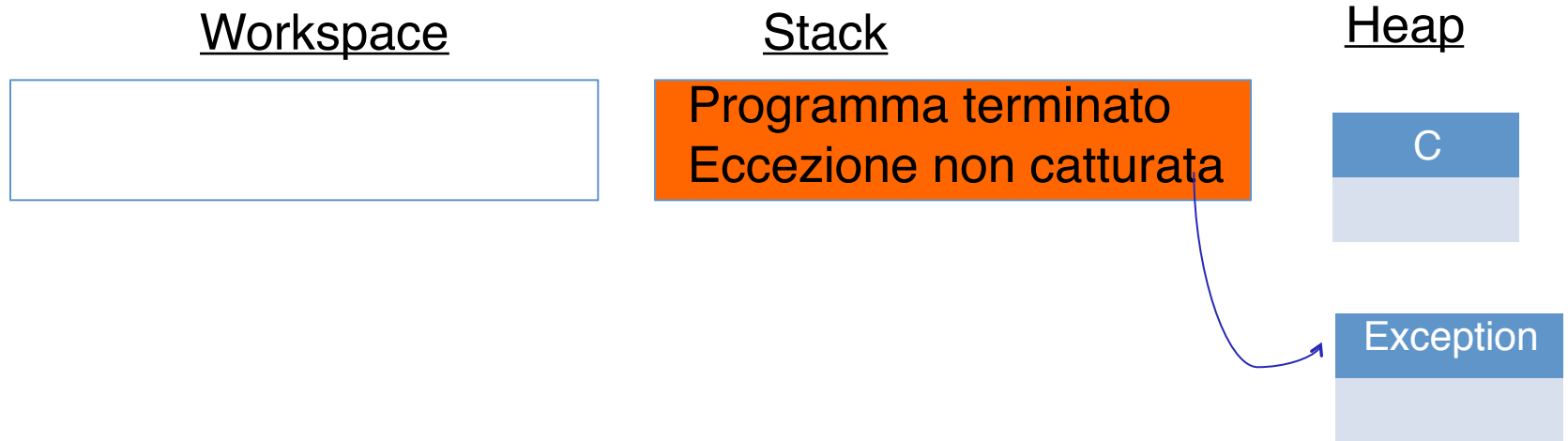


Elimina il codice corrente dal workspace
 Ricerca sullo stack il **try-catch** più recente
 In caso negativo, abort con errore

Abstract Stack Machine



Abstract Stack Machine



Un altro esempio

```
public class C {
    public void via(){
        primo();
        System.out.println("Sei al via");
    }

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

    public void secondo(){
        throw new Exception();
        System.out.println("Sei al secondo");
    }
}
```

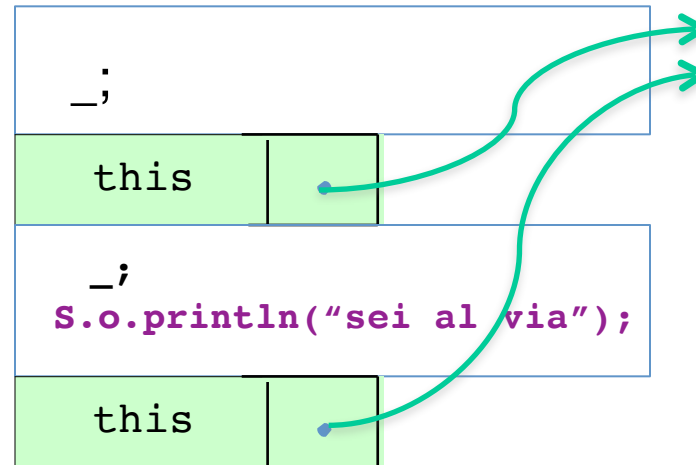
Cosa succede con `(new C()).via();`?

Abstract Stack Machine

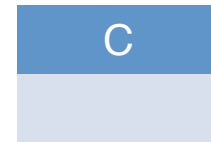
Workspace

```
try{secondo( ); }  
catch( ... ) { ... }  
S.o.println("sei al primo");
```

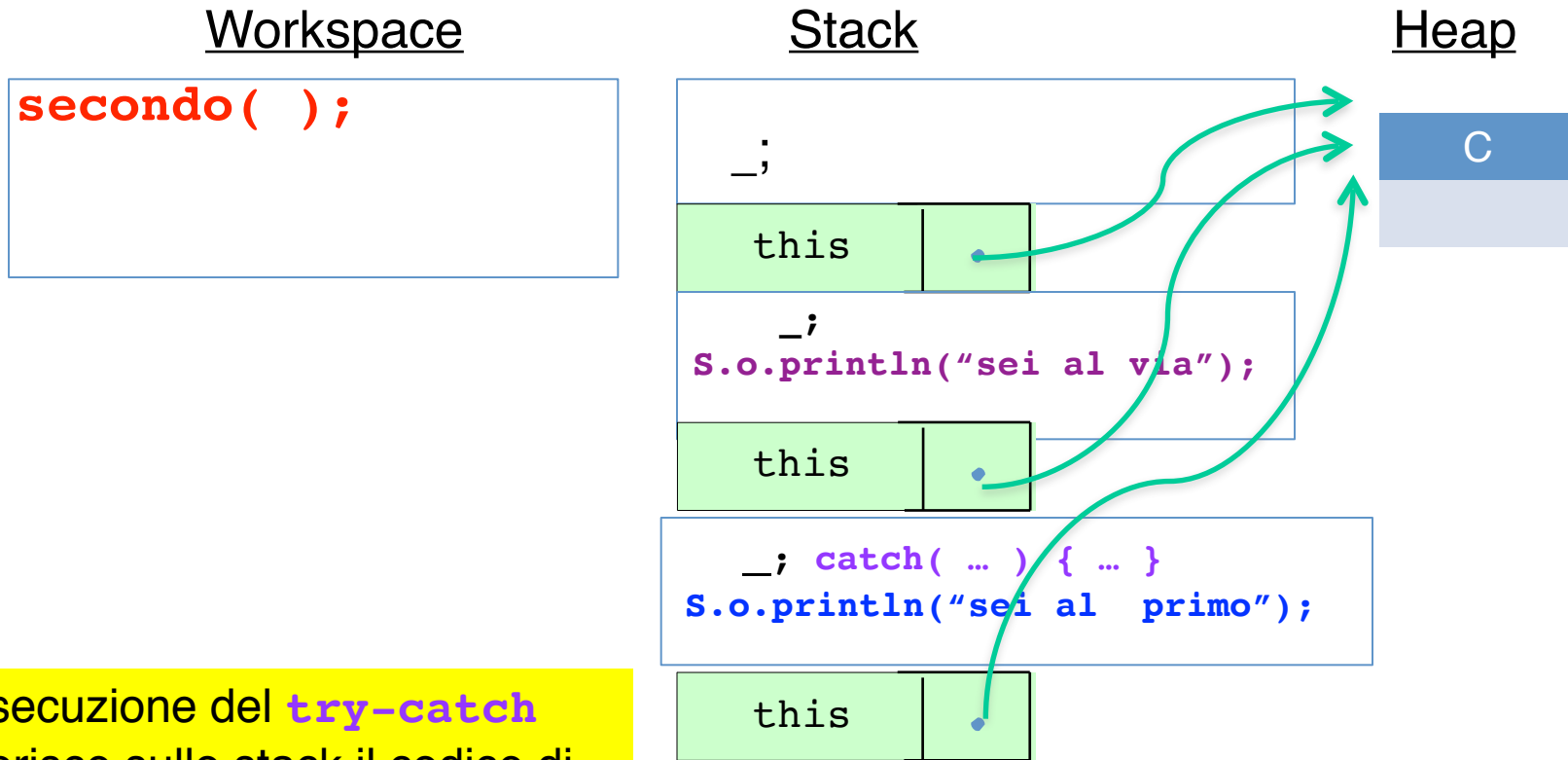
Stack



Heap



Abstract Stack Machine



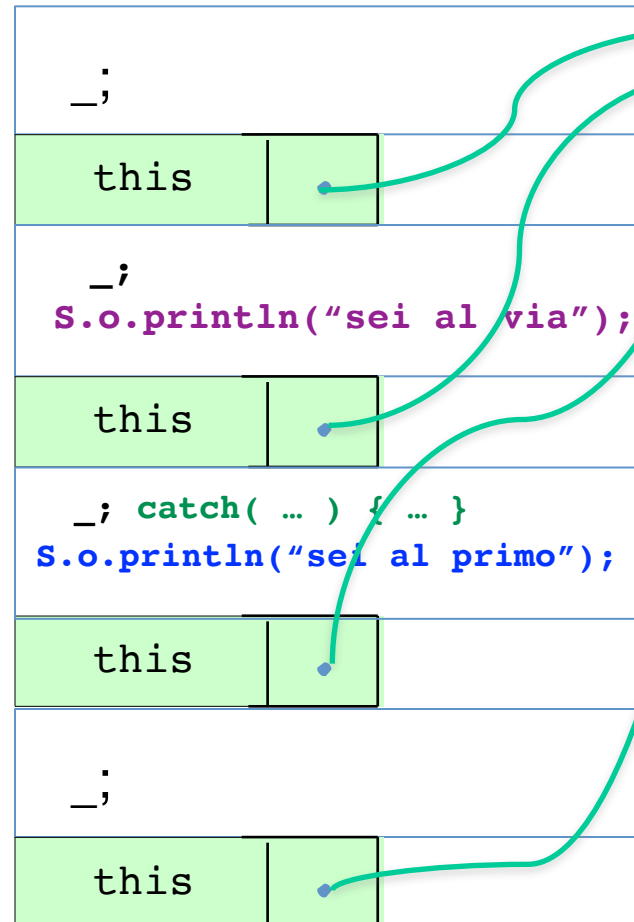
L'esecuzione del `try-catch` inserisce sullo stack il codice di gestione e lascia sul workspace il codice monitorato

Abstract Stack Machine

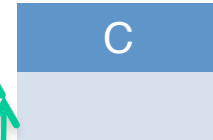
Workspace

```
throw new Exception();
S.o.println("sei al secondo");
```

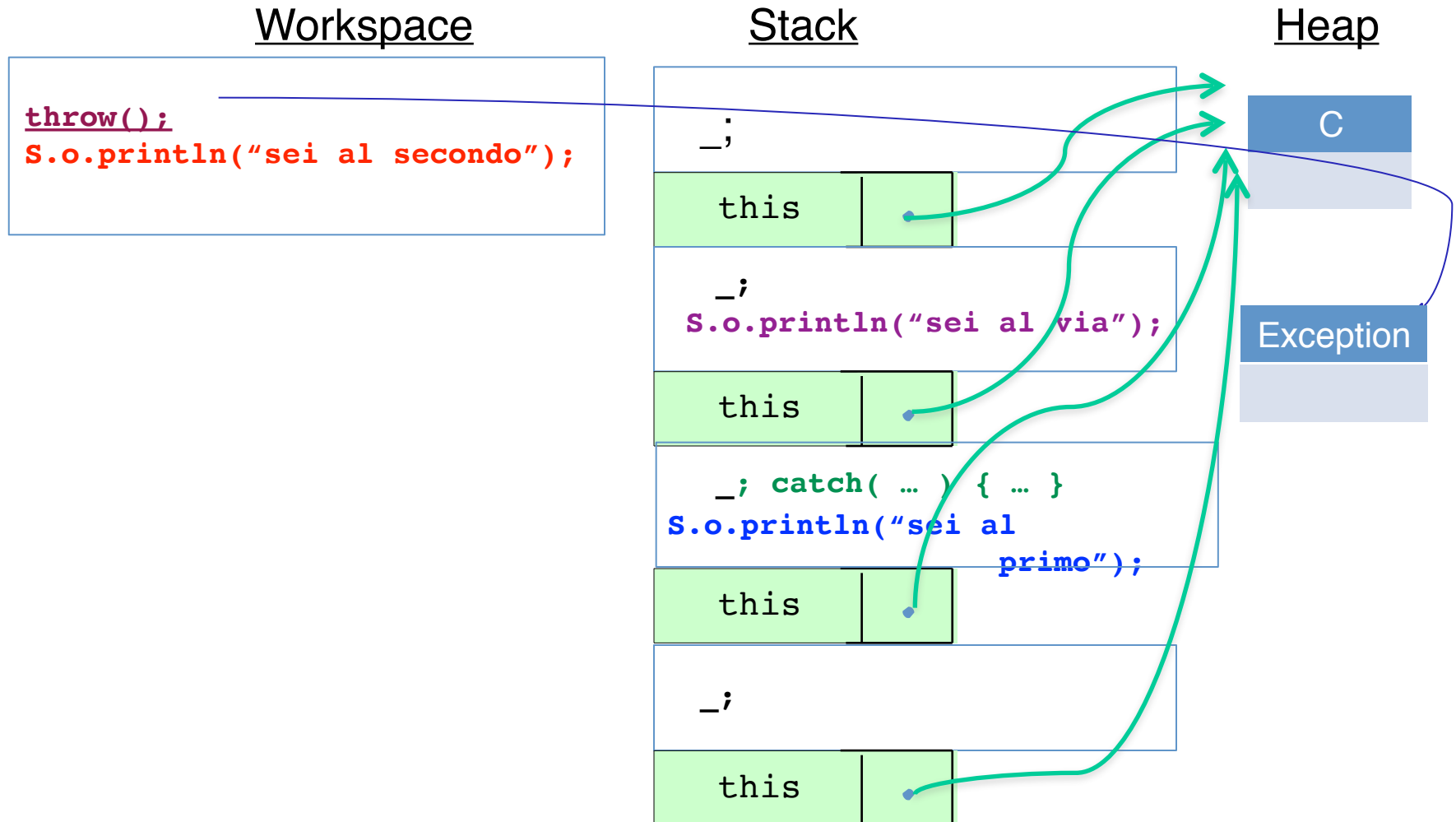
Stack



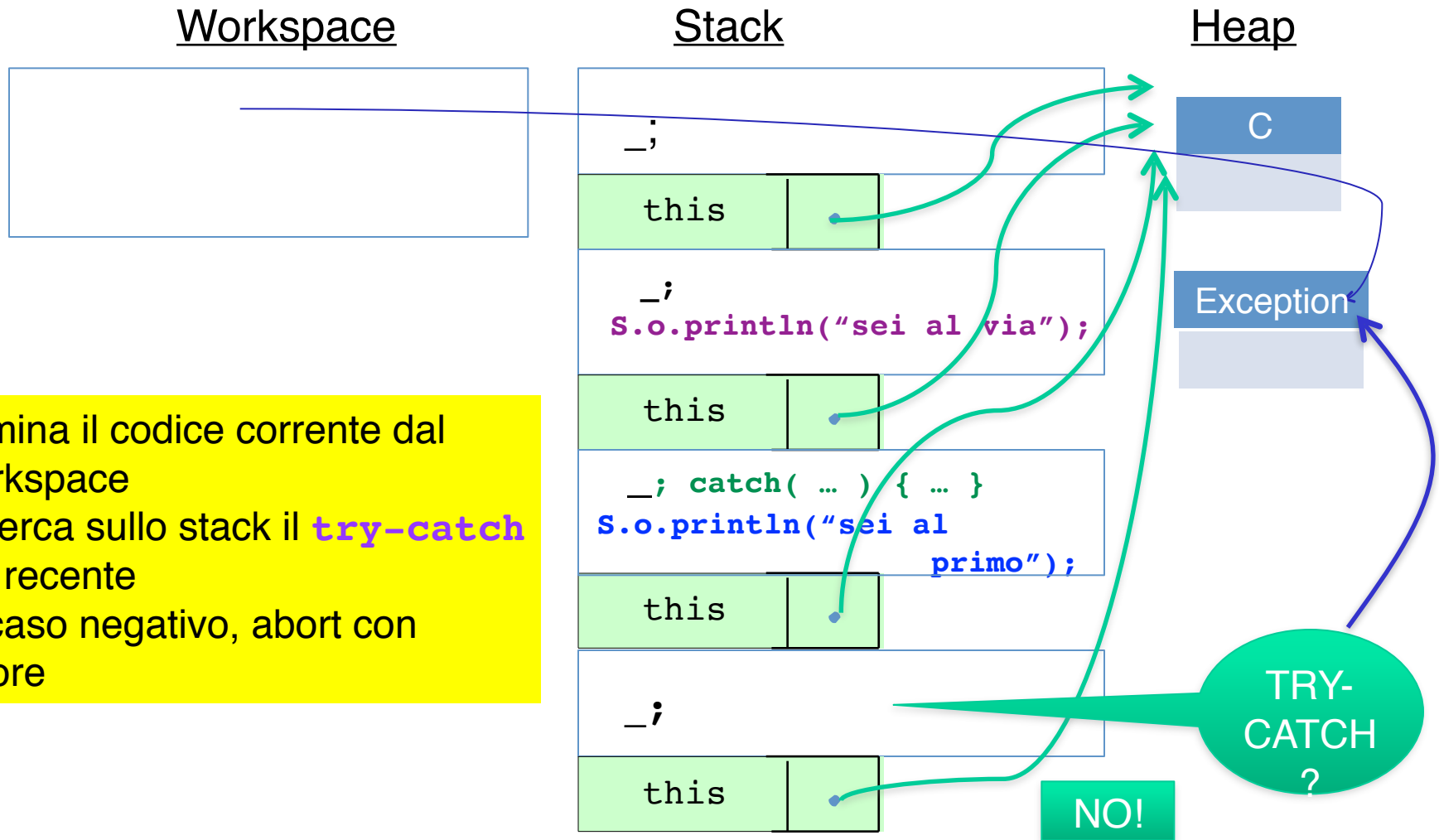
Heap



Abstract Stack Machine

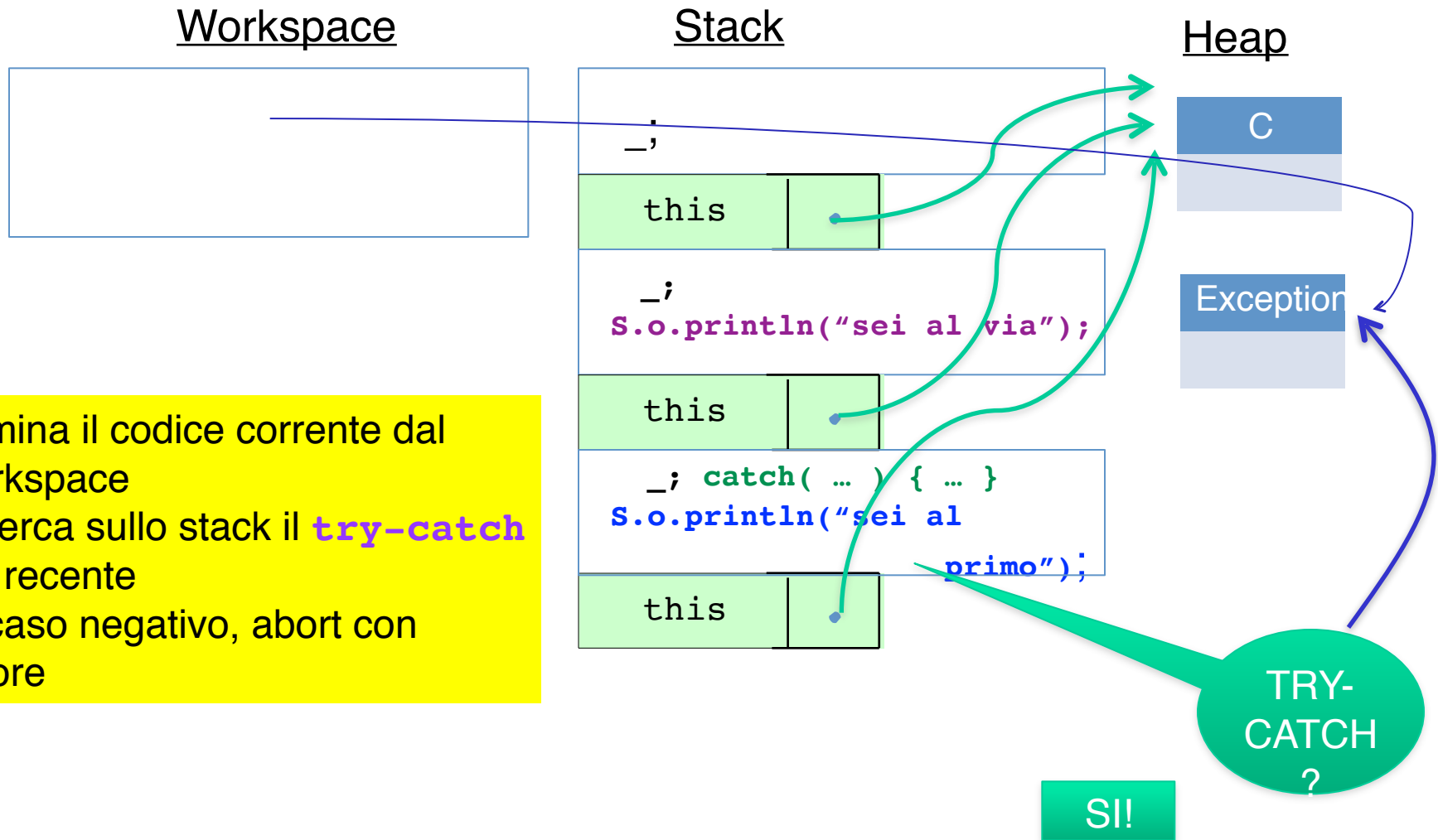


Abstract Stack Machine



Elimina il codice corrente dal workspace
 Ricerca sullo stack il **try-catch** più recente
 In caso negativo, abort con errore

Abstract Stack Machine

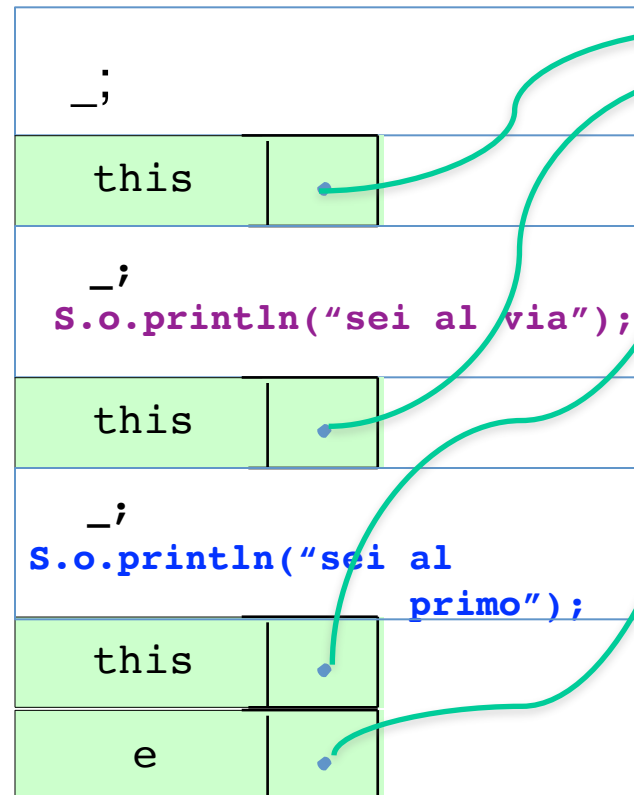


Abstract Stack Machine

Workspace

```
S.o.println
("catturata");
```

Stack



Heap

