test

The INFDEV team

Hogeschool Rotterdam
Rotterdam, Netherlands

```
1  interface ICounter {
2    void Incr(int diff);
3  }
4  class Counter : ICounter {
5    private int cnt;
6    public Counter() {
7      this.cnt = 0;
8    }
9    public void Incr(int diff) {
10     this.cnt = (this.cnt + diff);
11   }
12 }
13 ICounter c = new Counter();
14 c.Incr(5);
```

Stack:

PC
1

```
1  interface ICounter {
2    void Incr(int diff);
3  }
4  class Counter : ICounter {
5    private int cnt;
6    public Counter() {
7      this.cnt = 0;
8    }
9    public void Incr(int diff) {
10     this.cnt = (this.cnt + diff);
11   }
12 }
13 ICounter c = new Counter();
14 c.Incr(5);
```

Stack:

PC
13

Heap:

1
cnt=

```
1  interface ICounter {
2    void Incr(int diff);
3  }
4  class Counter : ICounter {
5    private int cnt;
6    public Counter() {
7      this.cnt = 0;
8    }
9    public void Incr(int diff) {
10     this.cnt = (this.cnt + diff);
11   }
12 }
13 ICounter c = new Counter();
14 c.Incr(5);
```

Stack:

PC	...		PC	ret	this
13	...		7	null	ref 1

Heap:

1
cnt=

```
1  interface ICounter {
2    void Incr(int diff);
3  }
4  class Counter : ICounter {
5    private int cnt;
6    public Counter() {
7      this.cnt = 0;
8    }
9    public void Incr(int diff) {
10     this.cnt = (this.cnt + diff);
11   }
12 }
13 ICounter c = new Counter();
14 c.Incr(5);
```

Stack:

PC	...		PC	ret
13	...		7	null

Heap:

1
cnt=0

```
1  interface ICounter {
2    void Incr(int diff);
3  }
4  class Counter : ICounter {
5    private int cnt;
6    public Counter() {
7      this.cnt = 0;
8    }
9    public void Incr(int diff) {
10     this.cnt = (this.cnt + diff);
11   }
12 }
13 ICounter c = new Counter();
14 c.Incr(5);
```

Stack:

PC	c
14	ref 1

Heap:

1
cnt=0

```
1  interface ICounter {
2    void Incr(int diff);
3  }
4  class Counter : ICounter {
5    private int cnt;
6    public Counter() {
7      this.cnt = 0;
8    }
9    public void Incr(int diff) {
10     this.cnt = (this.cnt + diff);
11   }
12 }
13 ICounter c = new Counter();
14 c.Incr(5);
```

Stack:

PC	...		PC	ret	diff	this
14	...		10	null	5	ref 1

Heap:

1
cnt=0

```
1  interface ICounter {
2    void Incr(int diff);
3  }
4  class Counter : ICounter {
5    private int cnt;
6    public Counter() {
7      this.cnt = 0;
8    }
9    public void Incr(int diff) {
10     this.cnt = (this.cnt + diff);
11   }
12 }
13 ICounter c = new Counter();
14 c.Incr(5);
```

Stack:

PC	...		PC	ret
14	...		10	null

Heap:

1
cnt=5

```
1  interface ICounter {
2    void Incr(int diff);
3  }
4  class Counter : ICounter {
5    private int cnt;
6    public Counter() {
7      this.cnt = 0;
8    }
9    public void Incr(int diff) {
10     this.cnt = (this.cnt + diff);
11   }
12 }
13 ICounter c = new Counter();
14 c.Incr(5);
```

Stack:

PC	c
15	ref 1

Heap:

1
cnt=5

```
1  interface ICounter {
2    void Incr(int diff);
3  }
4  class Counter : ICounter {
5    private int cnt;
6    public Counter() {
7      this.cnt = 0;
8    }
9    public void Incr(int diff) {
10     this.cnt = (this.cnt + diff);
11   }
12 }
13 ICounter c = new Counter();
14 c.Incr(5);
```

Declarations:

PC
1

```
1  interface ICounter {
2    void Incr(int diff);
3  }
4  class Counter : ICounter {
5    private int cnt;
6    public Counter() {
7      this.cnt = 0;
8    }
9    public void Incr(int diff) {
10     this.cnt = (this.cnt + diff);
11   }
12 }
13 ICounter c = new Counter();
14 c.Incr(5);
```

Declarations:

PC
4

Classes:

ICounter
Incr=int → void

```
1  interface ICounter {
2    void Incr(int diff);
3  }
4  class Counter : ICounter {
5    private int cnt;
6    public Counter() {
7      this.cnt = 0;
8    }
9    public void Incr(int diff) {
10     this.cnt = (this.cnt + diff);
11   }
12 }
13 ICounter c = new Counter();
14 c.Incr(5);
```

Declarations:

PC	this
7	Counter

Classes:

Counter	ICounter
Counter=Counter $\rightarrow$ Counter Incr=(Counter$\times$int) $\rightarrow$ void cnt=int	Incr=int $\rightarrow$ void

```
1  interface ICounter {
2    void Incr(int diff);
3  }
4  class Counter : ICounter {
5    private int cnt;
6    public Counter() {
7      this.cnt = 0;
8    }
9    public void Incr(int diff) {
10     this.cnt = (this.cnt + diff);
11   }
12 }
13 ICounter c = new Counter();
14 c.Incr(5);
```

Declarations:

PC	diff	this
9	int	Counter

Classes:

Counter	ICounter
Counter=Counter $\rightarrow$ Counter Incr=(Counter$\times$int) $\rightarrow$ void cnt=int	Incr=int $\rightarrow$ void

```
1  interface ICounter {
2    void Incr(int diff);
3  }
4  class Counter : ICounter {
5    private int cnt;
6    public Counter() {
7      this.cnt = 0;
8    }
9    public void Incr(int diff) {
10     this.cnt = (this.cnt + diff);
11   }
12 }
13 ICounter c = new Counter();
14 c.Incr(5);
```

Declarations:

PC
13

Classes:

Counter	ICounter
Counter=Counter → Counter Incr=(Counter×int) → void cnt=int	Incr=int → void

```
1  interface ICounter {
2    void Incr(int diff);
3  }
4  class Counter : ICounter {
5    private int cnt;
6    public Counter() {
7      this.cnt = 0;
8    }
9    public void Incr(int diff) {
10     this.cnt = (this.cnt + diff);
11   }
12 }
13 ICounter c = new Counter();
14 c.Incr(5);
```

Declarations:

PC	c
14	Counter

Classes:

Counter	ICounter
Counter=Counter → Counter Incr=(Counter×int) → void cnt=int	Incr=int → void

```
1  interface ICounter {
2    void Incr(int diff);
3  }
4  class Counter : ICounter {
5    private int cnt;
6    public Counter() {
7      this.cnt = 0;
8    }
9    public void Incr(int diff) {
10     this.cnt = (this.cnt + diff);
11   }
12 }
13 ICounter c = new Counter();
14 c.Incr(5);
```

Declarations:

c		PC	ret	arg_1	this
Counter		14	null	int	Counter

Classes:

Counter	ICounter
Counter=Counter → Counter Incr=(Counter×int) → void cnt=int	Incr=int → void

```
1  interface ICounter {
2    void Incr(int diff);
3  }
4  class Counter : ICounter {
5    private int cnt;
6    public Counter() {
7      this.cnt = 0;
8    }
9    public void Incr(int diff) {
10     this.cnt = (this.cnt + diff);
11   }
12 }
13 ICounter c = new Counter();
14 c.Incr(5);
```

Declarations:

c		PC	ret	arg_1	this
Counter		14	void	int	Counter

Classes:

Counter	ICounter
Counter=Counter $\rightarrow$ Counter Incr=(Counter$\times$int) $\rightarrow$ void cnt=int	Incr=int $\rightarrow$ void

```
1  interface ICounter {
2    void Incr(int diff);
3  }
4  class Counter : ICounter {
5    private int cnt;
6    public Counter() {
7      this.cnt = 0;
8    }
9    public void Incr(int diff) {
10     this.cnt = (this.cnt + diff);
11   }
12 }
13 ICounter c = new Counter();
14 c.Incr(5);
```

Declarations:

PC	c
15	Counter

Classes:

Counter	ICounter
Counter=Counter $\rightarrow$ Counter Incr=(Counter$\times$int) $\rightarrow$ void cnt=int	Incr=int $\rightarrow$ void

This is it!

The best of luck, and thanks for the attention!