

JVM Memory usage

Thread Stack

Heap memory

```
1  class Employee {
2      String name;
3      Integer salary;
4      Integer sales;
5      Integer bonus;
6
7      public Employee(String name, Integer salary, Integer sales) {
8          this.name = name;
9          this.salary = salary;
10         this.sales = sales;
11     }
12 }
13
14 public class Test {
15     static int BONUS_PERCENTAGE = 10;
16
17     static int getBonusPercentage(int salary) {
18         int percentage = salary * BONUS_PERCENTAGE / 100;
19         return percentage;
20     }
21
22     static int findEmployeeBonus(int salary, int noOfSales) {
23         int bonusPercentage = getBonusPercentage(salary);
24         int bonus = bonusPercentage * noOfSales;
25         return bonus;
26     }
27
28     public static void main(String[] args) {
29         Employee john = new Employee("John", 5000, 5);
30         john.bonus = findEmployeeBonus(john.salary, john.sales);
31         System.out.println(john.bonus);
32     }
33 }
```

JVM Memory usage

```
1  class Employee {
2      String name;
3      Integer salary;
4      Integer sales;
5      Integer bonus;
6
7      public Employee(String name, Integer salary, Integer sales) {
8          this.name = name;
9          this.salary = salary;
10         this.sales = sales;
11     }
12 }
13
14 public class Test {
15     static int BONUS_PERCENTAGE = 10;
16
17     static int getBonusPercentage(int salary) {
18         int percentage = salary * BONUS_PERCENTAGE / 100;
19         return percentage;
20     }
21
22     static int findEmployeeBonus(int salary, int noOfSales) {
23         int bonusPercentage = getBonusPercentage(salary);
24         int bonus = bonusPercentage * noOfSales;
25         return bonus;
26     }
27
28     public static void main(String[] args) {
29         Employee john = new Employee("John", 5000, 5);
30         john.bonus = findEmployeeBonus(john.salary, john.sales);
31         System.out.println(john.bonus);
32     }
33 }
```

Thread Stack

main

Heap memory

JVM Memory usage

```
1 class Employee {
2     String name;
3     Integer salary;
4     Integer sales;
5     Integer bonus;
6
7     public Employee(String name, Integer salary, Integer sales) {
8         this.name = name;
9         this.salary = salary;
10        this.sales = sales;
11    }
12 }
13
14 public class Test {
15     static int BONUS_PERCENTAGE = 10;
16
17     static int getBonusPercentage(int salary) {
18         int percentage = salary * BONUS_PERCENTAGE / 100;
19         return percentage;
20     }
21
22     static int findEmployeeBonus(int salary, int noOfSales) {
23         int bonusPercentage = getBonusPercentage(salary);
24         int bonus = bonusPercentage * noOfSales;
25         return bonus;
26     }
27
28     public static void main(String[] args) {
29         Employee john = new Employee("John", 5000, 5);
30         john.bonus = findEmployeeBonus(john.salary, john.sales);
31         System.out.println(john.bonus);
32     }
33 }
```

Thread Stack

Employee:new

this	●
name	●
salary	●
sales	●

main

Heap memory

Employee instance

name	null
salary	null
sales	null
bonus	null

String instance

"John"

Integer instance

5000

Integer instance

5

JVM Memory usage

```
1 class Employee {
2     String name;
3     Integer salary;
4     Integer sales;
5     Integer bonus;
6
7     public Employee(String name, Integer salary, Integer sales) {
8         this.name = name;
9         this.salary = salary;
10        this.sales = sales;
11    }
12 }
13
14 public class Test {
15     static int BONUS_PERCENTAGE = 10;
16
17     static int getBonusPercentage(int salary) {
18         int percentage = salary * BONUS_PERCENTAGE / 100;
19         return percentage;
20     }
21
22     static int findEmployeeBonus(int salary, int noOfSales) {
23         int bonusPercentage = getBonusPercentage(salary);
24         int bonus = bonusPercentage * noOfSales;
25         return bonus;
26     }
27
28     public static void main(String[] args) {
29         Employee john = new Employee("John", 5000, 5);
30         john.bonus = findEmployeeBonus(john.salary, john.sales);
31         System.out.println(john.bonus);
32     }
33 }
```

Thread Stack

Employee:new

this	●
name	●
salary	●
sales	●

main

Heap memory

Employee instance

name	null
salary	null
sales	null
bonus	null

String instance

"John"

Integer instance

5000

Integer instance

5

JVM Memory usage

```
1 class Employee {
2     String name;
3     Integer salary;
4     Integer sales;
5     Integer bonus;
6
7     public Employee(String name, Integer salary, Integer sales) {
8         this.name = name;
9         this.salary = salary;
10        this.sales = sales;
11    }
12 }
13
14 public class Test {
15     static int BONUS_PERCENTAGE = 10;
16
17     static int getBonusPercentage(int salary) {
18         int percentage = salary * BONUS_PERCENTAGE / 100;
19         return percentage;
20     }
21
22     static int findEmployeeBonus(int salary, int noOfSales) {
23         int bonusPercentage = getBonusPercentage(salary);
24         int bonus = bonusPercentage * noOfSales;
25         return bonus;
26     }
27
28     public static void main(String[] args) {
29         Employee john = new Employee("John", 5000, 5);
30         john.bonus = findEmployeeBonus(john.salary, john.sales);
31         System.out.println(john.bonus);
32     }
33 }
```

Thread Stack

Employee:new

this	●
name	●
salary	●
sales	●

main

Heap memory

Employee instance

name	●
salary	null
sales	null
bonus	null

String instance

"John"

Integer instance

5000

Integer instance

5

JVM Memory usage

```
1 class Employee {
2     String name;
3     Integer salary;
4     Integer sales;
5     Integer bonus;
6
7     public Employee(String name, Integer salary, Integer sales) {
8         this.name = name;
9         this.salary = salary;
10        this.sales = sales;
11    }
12 }
13
14 public class Test {
15     static int BONUS_PERCENTAGE = 10;
16
17     static int getBonusPercentage(int salary) {
18         int percentage = salary * BONUS_PERCENTAGE / 100;
19         return percentage;
20     }
21
22     static int findEmployeeBonus(int salary, int noOfSales) {
23         int bonusPercentage = getBonusPercentage(salary);
24         int bonus = bonusPercentage * noOfSales;
25         return bonus;
26     }
27
28     public static void main(String[] args) {
29         Employee john = new Employee("John", 5000, 5);
30         john.bonus = findEmployeeBonus(john.salary, john.sales);
31         System.out.println(john.bonus);
32     }
33 }
```

Thread Stack

Employee:new

this	●
name	●
salary	●
sales	●

main

Heap memory

Employee instance

name	●
salary	●
sales	null
bonus	null

String instance

"John"

Integer instance

5000

Integer instance

5

JVM Memory usage

```
1 class Employee {
2     String name;
3     Integer salary;
4     Integer sales;
5     Integer bonus;
6
7     public Employee(String name, Integer salary, Integer sales) {
8         this.name = name;
9         this.salary = salary;
10        this.sales = sales;
11    }
12 }
13
14 public class Test {
15     static int BONUS_PERCENTAGE = 10;
16
17     static int getBonusPercentage(int salary) {
18         int percentage = salary * BONUS_PERCENTAGE / 100;
19         return percentage;
20     }
21
22     static int findEmployeeBonus(int salary, int noOfSales) {
23         int bonusPercentage = getBonusPercentage(salary);
24         int bonus = bonusPercentage * noOfSales;
25         return bonus;
26     }
27
28     public static void main(String[] args) {
29         Employee john = new Employee("John", 5000, 5);
30         john.bonus = findEmployeeBonus(john.salary, john.sales);
31         System.out.println(john.bonus);
32     }
33 }
```

Thread Stack

Employee:new

this	●
name	●
salary	●
sales	●

main

Heap memory

Employee instance

name	●
salary	●
sales	●
bonus	null

String instance

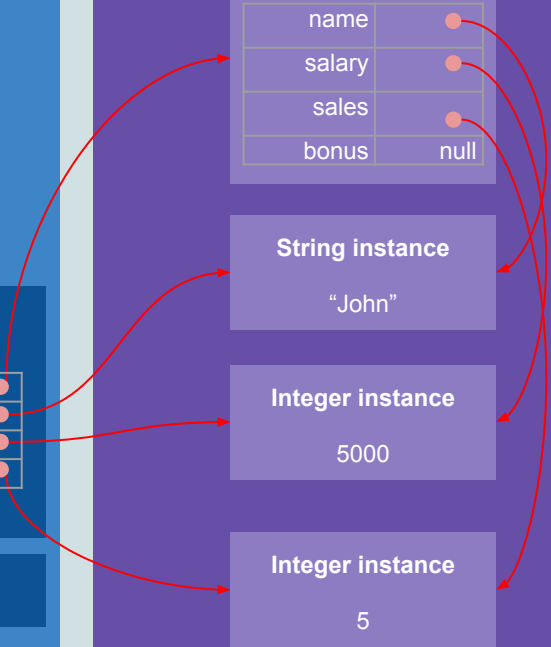
"John"

Integer instance

5000

Integer instance

5



JVM Memory usage

```
1  class Employee {
2      String name;
3      Integer salary;
4      Integer sales;
5      Integer bonus;
6
7      public Employee(String name, Integer salary, Integer sales) {
8          this.name = name;
9          this.salary = salary;
10         this.sales = sales;
11     }
12 }
13
14 public class Test {
15     static int BONUS_PERCENTAGE = 10;
16
17     static int getBonusPercentage(int salary) {
18         int percentage = salary * BONUS_PERCENTAGE / 100;
19         return percentage;
20     }
21
22     static int findEmployeeBonus(int salary, int noOfSales) {
23         int bonusPercentage = getBonusPercentage(salary);
24         int bonus = bonusPercentage * noOfSales;
25         return bonus;
26     }
27
28     public static void main(String[] args) {
29         Employee john = new Employee("John", 5000, 5);
30         john.bonus = findEmployeeBonus(john.salary, john.sales);
31         System.out.println(john.bonus);
32     }
33 }
```

Thread Stack

Employee:new

this	●
name	●
salary	●
sales	●
return	void

main

Heap memory

Employee instance

name	●
salary	●
sales	●
bonus	null

String instance

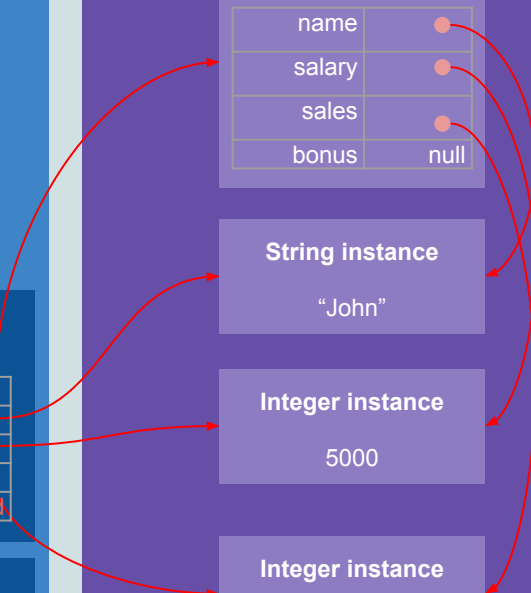
"John"

Integer instance

5000

Integer instance

5



JVM Memory usage

```
1 class Employee {
2     String name;
3     Integer salary;
4     Integer sales;
5     Integer bonus;
6
7     public Employee(String name, Integer salary, Integer sales) {
8         this.name = name;
9         this.salary = salary;
10        this.sales = sales;
11    }
12 }
13
14 public class Test {
15     static int BONUS_PERCENTAGE = 10;
16
17     static int getBonusPercentage(int salary) {
18         int percentage = salary * BONUS_PERCENTAGE / 100;
19         return percentage;
20     }
21
22     static int findEmployeeBonus(int salary, int noOfSales) {
23         int bonusPercentage = getBonusPercentage(salary);
24         int bonus = bonusPercentage * noOfSales;
25         return bonus;
26     }
27
28     public static void main(String[] args) {
29         Employee john = new Employee("John", 5000, 5);
30         john.bonus = findEmployeeBonus(john.salary, john.sales);
31         System.out.println(john.bonus);
32     }
33 }
```

Thread Stack

main

john

Heap memory

Employee instance

name	●
salary	●
sales	●
bonus	null

String instance

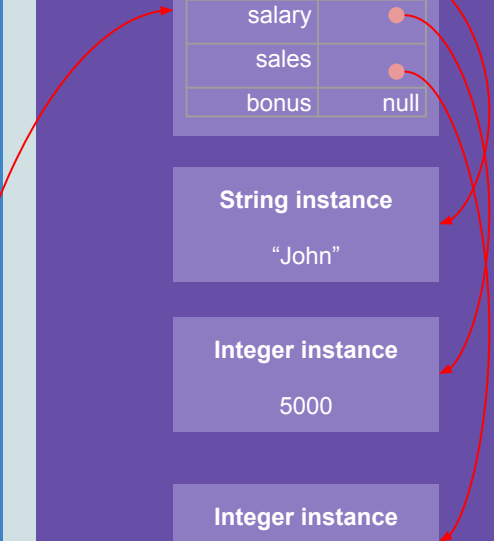
"John"

Integer instance

5000

Integer instance

5



JVM Memory usage

```
1  class Employee {
2      String name;
3      Integer salary;
4      Integer sales;
5      Integer bonus;
6
7      public Employee(String name, Integer salary, Integer sales) {
8          this.name = name;
9          this.salary = salary;
10         this.sales = sales;
11     }
12 }
13
14 public class Test {
15     static int BONUS_PERCENTAGE = 10;
16
17     static int getBonusPercentage(int salary) {
18         int percentage = salary * BONUS_PERCENTAGE / 100;
19         return percentage;
20     }
21
22     static int findEmployeeBonus(int salary, int noOfSales) {
23         int bonusPercentage = getBonusPercentage(salary);
24         int bonus = bonusPercentage * noOfSales;
25         return bonus;
26     }
27
28     public static void main(String[] args) {
29         Employee john = new Employee("John", 5000, 5);
30         john.bonus = findEmployeeBonus(john.salary, john.sales);
31         System.out.println(john.bonus);
32     }
33 }
```

Thread Stack

findEmployeeBonus

salary	5000
noOfSales	5

main

john	●
------	---

Heap memory

Employee instance

name	●
salary	●
sales	●
bonus	null

String instance

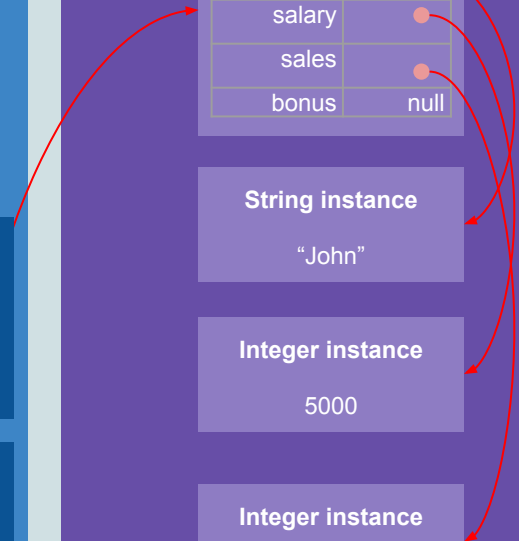
"John"

Integer instance

5000

Integer instance

5



JVM Memory usage

```
1  class Employee {
2      String name;
3      Integer salary;
4      Integer sales;
5      Integer bonus;
6
7      public Employee(String name, Integer salary, Integer sales) {
8          this.name = name;
9          this.salary = salary;
10         this.sales = sales;
11     }
12 }
13
14 public class Test {
15     static int BONUS_PERCENTAGE = 10;
16
17     static int getBonusPercentage(int salary) {
18         int percentage = salary * BONUS_PERCENTAGE / 100;
19         return percentage;
20     }
21
22     static int findEmployeeBonus(int salary, int noOfSales) {
23         int bonusPercentage = getBonusPercentage(salary);
24         int bonus = bonusPercentage * noOfSales;
25         return bonus;
26     }
27
28     public static void main(String[] args) {
29         Employee john = new Employee("John", 5000, 5);
30         john.bonus = findEmployeeBonus(john.salary, john.sales);
31         System.out.println(john.bonus);
32     }
33 }
```

Thread Stack

getBonusPercentage

salary	5000
--------	------

findEmployeeBonus

salary	5000
noOfSales	5

main

john	●
------	---

Heap memory

Employee instance

name	●
salary	●
sales	●
bonus	null

String instance

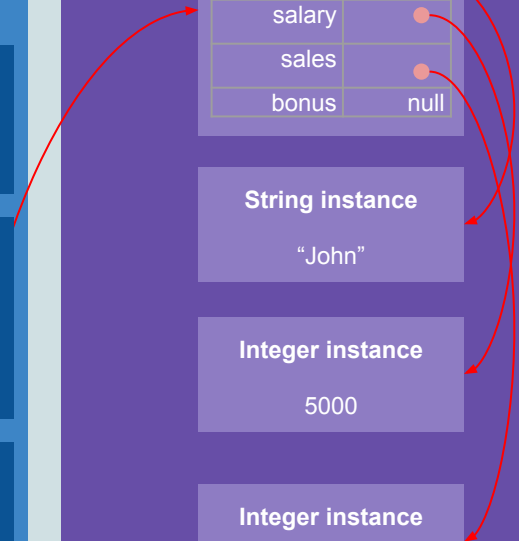
"John"

Integer instance

5000

Integer instance

5



JVM Memory usage

```
1  class Employee {
2      String name;
3      Integer salary;
4      Integer sales;
5      Integer bonus;
6
7      public Employee(String name, Integer salary, Integer sales) {
8          this.name = name;
9          this.salary = salary;
10         this.sales = sales;
11     }
12 }
13
14 public class Test {
15     static int BONUS_PERCENTAGE = 10;
16
17     static int getBonusPercentage(int salary) {
18         int percentage = salary * BONUS_PERCENTAGE / 100;
19         return percentage;
20     }
21
22     static int findEmployeeBonus(int salary, int noOfSales) {
23         int bonusPercentage = getBonusPercentage(salary);
24         int bonus = bonusPercentage * noOfSales;
25         return bonus;
26     }
27
28     public static void main(String[] args) {
29         Employee john = new Employee("John", 5000, 5);
30         john.bonus = findEmployeeBonus(john.salary, john.sales);
31         System.out.println(john.bonus);
32     }
33 }
```

Thread Stack

getBonusPercentage

salary	5000
percentage	500

findEmployeeBonus

salary	5000
noOfSales	5

main

john	●
------	---

Heap memory

Employee instance

name	●
salary	●
sales	●
bonus	null

String instance

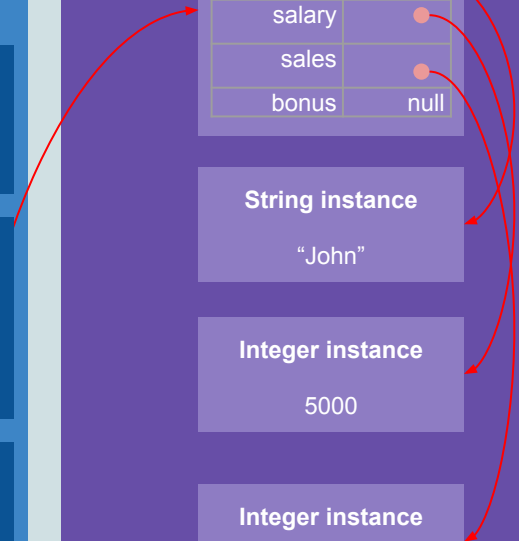
"John"

Integer instance

5000

Integer instance

5



JVM Memory usage

```
1 class Employee {
2     String name;
3     Integer salary;
4     Integer sales;
5     Integer bonus;
6
7     public Employee(String name, Integer salary, Integer sales) {
8         this.name = name;
9         this.salary = salary;
10        this.sales = sales;
11    }
12 }
13
14 public class Test {
15     static int BONUS_PERCENTAGE = 10;
16
17     static int getBonusPercentage(int salary) {
18         int percentage = salary * BONUS_PERCENTAGE / 100;
19         return percentage;
20     }
21
22     static int findEmployeeBonus(int salary, int noOfSales) {
23         int bonusPercentage = getBonusPercentage(salary);
24         int bonus = bonusPercentage * noOfSales;
25         return bonus;
26     }
27
28     public static void main(String[] args) {
29         Employee john = new Employee("John", 5000, 5);
30         john.bonus = findEmployeeBonus(john.salary, john.sales);
31         System.out.println(john.bonus);
32     }
33 }
```

Thread Stack

getBonusPercentage

salary	5000
percentage	500
return	500

findEmployeeBonus

salary	5000
noOfSales	5

main

john	●
------	---

Heap memory

Employee instance

name	●
salary	●
sales	●
bonus	null

String instance

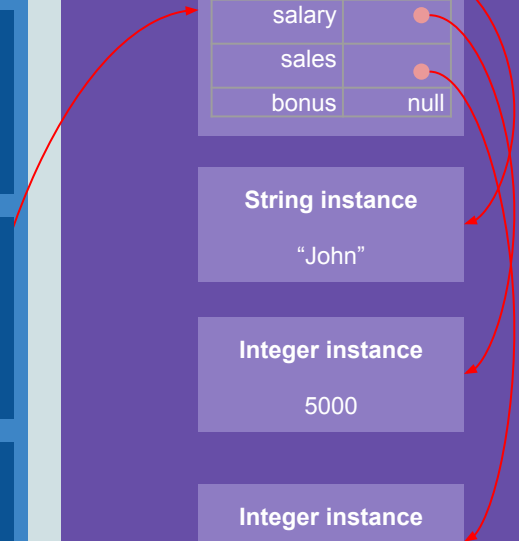
"John"

Integer instance

5000

Integer instance

5



JVM Memory usage

```
1 class Employee {
2     String name;
3     Integer salary;
4     Integer sales;
5     Integer bonus;
6
7     public Employee(String name, Integer salary, Integer sales) {
8         this.name = name;
9         this.salary = salary;
10        this.sales = sales;
11    }
12 }
13
14 public class Test {
15     static int BONUS_PERCENTAGE = 10;
16
17     static int getBonusPercentage(int salary) {
18         int percentage = salary * BONUS_PERCENTAGE / 100;
19         return percentage;
20     }
21
22     static int findEmployeeBonus(int salary, int noOfSales) {
23         int bonusPercentage = getBonusPercentage(salary);
24         int bonus = bonusPercentage * noOfSales;
25         return bonus;
26     }
27
28     public static void main(String[] args) {
29         Employee john = new Employee("John", 5000, 5);
30         john.bonus = findEmployeeBonus(john.salary, john.sales);
31         System.out.println(john.bonus);
32     }
33 }
```

Thread Stack

findEmployeeBonus

salary	5000
noOfSales	5
bonusPercentage	500
bonus	2500

main

john	●
------	---

Heap memory

Employee instance

name	●
salary	●
sales	●
bonus	null

String instance

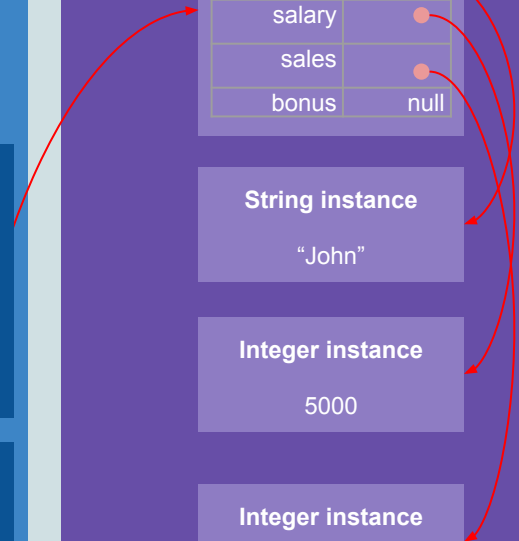
"John"

Integer instance

5000

Integer instance

5



JVM Memory usage

```
1 class Employee {
2     String name;
3     Integer salary;
4     Integer sales;
5     Integer bonus;
6
7     public Employee(String name, Integer salary, Integer sales) {
8         this.name = name;
9         this.salary = salary;
10        this.sales = sales;
11    }
12 }
13
14 public class Test {
15     static int BONUS_PERCENTAGE = 10;
16
17     static int getBonusPercentage(int salary) {
18         int percentage = salary * BONUS_PERCENTAGE / 100;
19         return percentage;
20     }
21
22     static int findEmployeeBonus(int salary, int noOfSales) {
23         int bonusPercentage = getBonusPercentage(salary);
24         int bonus = bonusPercentage * noOfSales;
25         return bonus;
26     }
27
28     public static void main(String[] args) {
29         Employee john = new Employee("John", 5000, 5);
30         john.bonus = findEmployeeBonus(john.salary, john.sales);
31         System.out.println(john.bonus);
32     }
33 }
```

Thread Stack

findEmployeeBonus

salary	5000
noOfSales	5
bonusPercentage	500
bonus	2500
return	2500

main

john	●
------	---

Heap memory

Employee instance

name	●
salary	●
sales	●
bonus	null

String instance

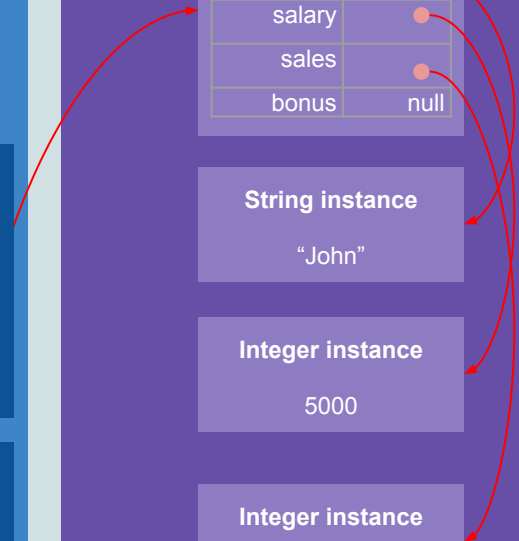
"John"

Integer instance

5000

Integer instance

5



JVM Memory usage

```
1 class Employee {
2     String name;
3     Integer salary;
4     Integer sales;
5     Integer bonus;
6
7     public Employee(String name, Integer salary, Integer sales) {
8         this.name = name;
9         this.salary = salary;
10        this.sales = sales;
11    }
12 }
13
14 public class Test {
15     static int BONUS_PERCENTAGE = 10;
16
17     static int getBonusPercentage(int salary) {
18         int percentage = salary * BONUS_PERCENTAGE / 100;
19         return percentage;
20     }
21
22     static int findEmployeeBonus(int salary, int noOfSales) {
23         int bonusPercentage = getBonusPercentage(salary);
24         int bonus = bonusPercentage * noOfSales;
25         return bonus;
26     }
27
28     public static void main(String[] args) {
29         Employee john = new Employee("John", 5000, 5);
30         john.bonus = findEmployeeBonus(john.salary, john.sales);
31         System.out.println(john.bonus);
32     }
33 }
```

Thread Stack

main

john

Heap memory

Employee instance

name

salary

sales

bonus

String instance

"John"

Integer instance

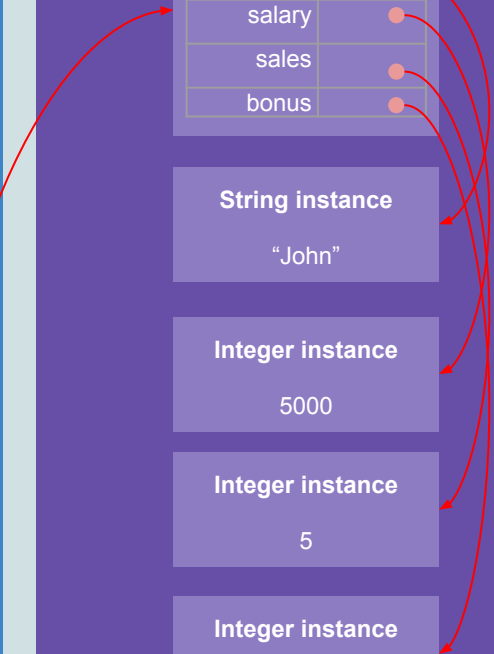
5000

Integer instance

5

Integer instance

2500



JVM Memory usage

```
1 class Employee {
2     String name;
3     Integer salary;
4     Integer sales;
5     Integer bonus;
6
7     public Employee(String name, Integer salary, Integer sales) {
8         this.name = name;
9         this.salary = salary;
10        this.sales = sales;
11    }
12 }
13
14 public class Test {
15     static int BONUS_PERCENTAGE = 10;
16
17     static int getBonusPercentage(int salary) {
18         int percentage = salary * BONUS_PERCENTAGE / 100;
19         return percentage;
20     }
21
22     static int findEmployeeBonus(int salary, int noOfSales) {
23         int bonusPercentage = getBonusPercentage(salary);
24         int bonus = bonusPercentage * noOfSales;
25         return bonus;
26     }
27
28     public static void main(String[] args) {
29         Employee john = new Employee("John", 5000, 5);
30         john.bonus = findEmployeeBonus(john.salary, john.sales);
31         System.out.println(john.bonus);
32     }
33 }
```

Thread Stack

main

john	•
return	void

Heap memory

Employee instance

name	•
salary	•
sales	•
bonus	•

String instance

"John"

Integer instance

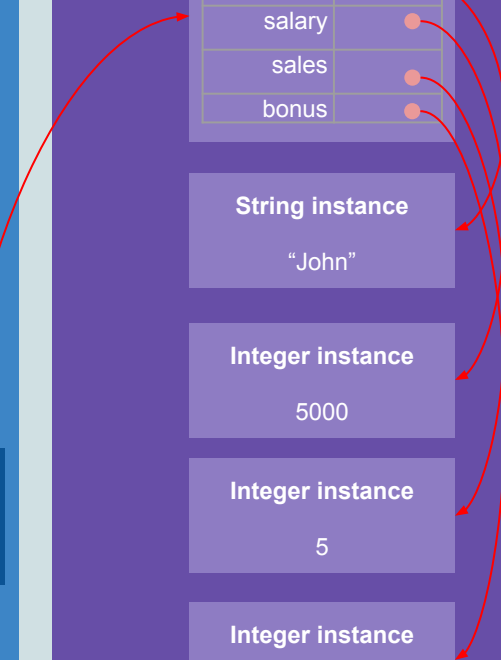
5000

Integer instance

5

Integer instance

2500



JVM Memory usage

```
1  class Employee {
2      String name;
3      Integer salary;
4      Integer sales;
5      Integer bonus;
6
7      public Employee(String name, Integer salary, Integer sales) {
8          this.name = name;
9          this.salary = salary;
10         this.sales = sales;
11     }
12 }
13
14 public class Test {
15     static int BONUS_PERCENTAGE = 10;
16
17     static int getBonusPercentage(int salary) {
18         int percentage = salary * BONUS_PERCENTAGE / 100;
19         return percentage;
20     }
21
22     static int findEmployeeBonus(int salary, int noOfSales) {
23         int bonusPercentage = getBonusPercentage(salary);
24         int bonus = bonusPercentage * noOfSales;
25         return bonus;
26     }
27
28     public static void main(String[] args) {
29         Employee john = new Employee("John", 5000, 5);
30         john.bonus = findEmployeeBonus(john.salary, john.sales);
31         System.out.println(john.bonus);
32     }
33 }
```

Thread Stack

Heap memory

Employee instance

name	●
salary	●
sales	●
bonus	●

String instance

"John"

Integer instance

5000

Integer instance

5

Integer instance

2500

