PARCIAL III - CI3641 - JOAO PINTO 17-10490

PREGUNTA 3

ASOCIACIÓN DINÁMICA DE MÉTODOS

EJECUCIÓN PASO A PASO

ASOCIACIÓN DINÁMICA DE MÉTODOS - EJECUCIÓN PASO A PASO

```
       class A {
           int a = 4, b = 9

           fun p(int x): int {
 0             a = b * x
 1             return q(a)
           }

           fun q(int y): int {
 2             return a + y
           }
       }

       class B extends A {
           A sopa = new C()

           fun q(int y): int {
 3             return sopa.p(a + b) + y
           }
       }

       class C extends B {
           int c = 0

           fun p(int x): int {
 4             a = 3 + x
 5             c = 2 * b − x
 6             return q(a + b + c)
           }

           fun q(int y): int {
 7             return c + y
           }
       }

 8     A mon = new B()
 9     A don = new C()
10     B go = new C()

11     print(mon.p(1) + don.p(1) + go.p(1))
```

Nombre	TE	TD	Methods table	Values
mon	A	B	p -> A, q -> B	a = 4, b = 9
don	A	C	p -> C, q -> C	a = 4, b = 9, c = 0
go	B	C	p -> C, q -> C	a = 4, b = 9, c = 0
mon.sopa	A	C	p -> C, q -> C	a = 4, b = 9, c = 0
don.sopa	A	C	p -> C, q -> C	a = 4, b = 9, c = 0
go.sopa	A	C	p -> C, q -> C	a = 4, b = 9, c = 0

Global	PC	11

ASOCIACIÓN DINÁMICA DE MÉTODOS - EJECUCIÓN PASO A PASO

```
       class A {
           int a = 4, b = 9

           fun p(int x): int {
 0             a = b * x
 1             return q(a)
           }

           fun q(int y): int {
 2             return a + y
           }
       }

       class B extends A {
           A sopa = new C()

           fun q(int y): int {
 3             return sopa.p(a + b) + y
           }
       }

       class C extends B {
           int c = 0

           fun p(int x): int {
 4             a = 3 + x
 5             c = 2 * b - x
 6             return q(a + b + c)
           }

           fun q(int y): int {
 7             return c + y
           }
       }

 8     A mon = new B()
 9     A don = new C()
10     B go = new C()

11     print(mon.p(1) + don.p(1) + go.p(1))
```

Nombre	TE	TD	Methods table	Values
mon	A	B	p -> A, q -> B	a = 9, b = 9
don	A	C	p -> C, q -> C	a = 4, b = 9, c = 0
go	B	C	p -> C, q -> C	a = 4, b = 9, c = 0
mon.sopa	A	C	p -> C, q -> C	a = 4, b = 9, c = 0
don.sopa	A	C	p -> C, q -> C	a = 4, b = 9, c = 0
go.sopa	A	C	p -> C, q -> C	a = 4, b = 9, c = 0

mon.p	x	1
	PC	1
	return	-

Global	PC	11

ASOCIACIÓN DINÁMICA DE MÉTODOS - EJECUCIÓN PASO A PASO

```
      class A {
          int a = 4, b = 9

          fun p(int x): int {
▸  0          a = b * x
▸  1          return q(a)
          }

          fun q(int y): int {
▸  2          return a + y
          }
      }

      class B extends A {
          A sopa = new C()

          fun q(int y): int {
▸  3          return sopa.p(a + b) + y
          }
      }

      class C extends B {
          int c = 0

          fun p(int x): int {
▸  4          a = 3 + x
▸  5          c = 2 * b − x
▸  6          return q(a + b + c)
          }

          fun q(int y): int {
▸  7          return c + y
          }
      }

▸  8  A mon = new B()
▸  9  A don = new C()
▸ 10  B go = new C()

▸ 11  print(mon.p(1) + don.p(1) + go.p(1))
```

Nombre	TE	TD	Methods table	Values
mon	A	B	p -> A, q -> B	a = 9, b = 9
don	A	C	p -> C, q -> C	a = 4, b = 9, c = 0
go	B	C	p -> C, q -> C	a = 4, b = 9, c = 0
mon.sopa	A	C	p -> C, q -> C	a = 4, b = 9, c = 0
don.sopa	A	C	p -> C, q -> C	a = 4, b = 9, c = 0
go.sopa	A	C	p -> C, q -> C	a = 4, b = 9, c = 0

mon.q	y	9
	PC	3
	return	-

mon.p	x	1
	PC	1
	return	-

Global	PC	11

ASOCIACIÓN DINÁMICA DE MÉTODOS - EJECUCIÓN PASO A PASO

```
       class A {
           int a = 4, b = 9

           fun p(int x): int {
 ▸ 0           a = b * x
 ▸ 1           return q(a)
           }

           fun q(int y): int {
 ▸ 2           return a + y
           }
       }

       class B extends A {
           A sopa = new C()

           fun q(int y): int {
 ▸ 3           return sopa.p(a + b) + y
           }
       }

       class C extends B {
           int c = 0

           fun p(int x): int {
 ▸ 4           a = 3 + x
 ▸ 5           c = 2 * b − x
 ▸ 6           return q(a + b + c)
           }

           fun q(int y): int {
 ▸ 7           return c + y
           }
       }

 ▸ 8   A mon = new B()
 ▸ 9   A don = new C()
 ▸ 10  B go = new C()

 ▸ 11  print(mon.p(1) + don.p(1) + go.p(1))
```

Nombre	TE	TD	Methods table	Values
mon	A	B	p -> A, q -> B	a = 9, b = 9
don	A	C	p -> C, q -> C	a = 4, b = 9, c = 0
go	B	C	p -> C, q -> C	a = 4, b = 9, c = 0
mon.sopa	A	C	p -> C, q -> C	a = 3 + 18, b = 9, c = 18 - 18
don.sopa	A	C	p -> C, q -> C	a = 4, b = 9, c = 0
go.sopa	A	C	p -> C, q -> C	a = 4, b = 9, c = 0

mon.sopa.p		
	x	18
	PC	6
	return	-

mon.q		
	y	9
	PC	3
	return	-

mon.p		
	x	1
	PC	1
	return	-

ASOCIACIÓN DINÁMICA DE MÉTODOS - EJECUCIÓN PASO A PASO

```
        class A {
            int a = 4, b = 9

            fun p(int x): int {
▸  0            a = b * x
▸  1            return q(a)
            }

            fun q(int y): int {
▸  2            return a + y
            }
        }

        class B extends A {
            A sopa = new C()

            fun q(int y): int {
▸  3            return sopa.p(a + b) + y
            }
        }

        class C extends B {
            int c = 0

            fun p(int x): int {
▸  4            a = 3 + x
▸  5            c = 2 * b − x
▸  6            return q(a + b + c)
            }

            fun q(int y): int {
▸  7            return c + y
            }
        }

▸  8    A mon = new B()
▸  9    A don = new C()
▸ 10    B go = new C()

▸ 11    print(mon.p(1) + don.p(1) + go.p(1))
```

Nombre	TE	TD	Methods table	Values
mon	A	B	p -> A, q -> B	a = 9, b = 9
don	A	C	p -> C, q -> C	a = 4, b = 9, c = 0
go	B	C	p -> C, q -> C	a = 4, b = 9, c = 0
mon.sopa	A	C	p -> C, q -> C	a = 21, b = 9, c = 0
don.sopa	A	C	p -> C, q -> C	a = 4, b = 9, c = 0
go.sopa	A	C	p -> C, q -> C	a = 4, b = 9, c = 0

mon.sopa.p	x	18
	PC	6
	return	-

mon.q	y	9
	PC	3
	return	-

mon.p	x	1
	PC	1
	return	-

ASOCIACIÓN DINÁMICA DE MÉTODOS - EJECUCIÓN PASO A PASO

```
     class A {
         int a = 4, b = 9

         fun p(int x): int {
 0           a = b * x
 1           return q(a)
         }

         fun q(int y): int {
 2           return a + y
         }
     }

     class B extends A {
         A sopa = new C()

         fun q(int y): int {
 3           return sopa.p(a + b) + y
         }
     }

     class C extends B {
         int c = 0

         fun p(int x): int {
 4           a = 3 + x
 5           c = 2 * b - x
 6           return q(a + b + c)
         }

         fun q(int y): int {
 7           return c + y
         }
     }

 8   A mon = new B()
 9   A don = new C()
10   B go = new C()

11   print(mon.p(1) + don.p(1) + go.p(1))
```

Nombre	TE	TD	Methods table	Values
mon	A	B	p -> A, q -> B	a = 9, b = 9
don	A	C	p -> C, q -> C	a = 4, b = 9, c = 0
go	B	C	p -> C, q -> C	a = 4, b = 9, c = 0
mon.sopa	A	C	p -> C, q -> C	a = 21, b = 9, c = 0
don.sopa	A	C	p -> C, q -> C	a = 4, b = 9, c = 0
go.sopa	A	C	p -> C, q -> C	a = 4, b = 9, c = 0

mon.sopa.q	y	30
	PC	7
	return	0 + 30 = 30

mon.sopa.p	x	18
	PC	6
	return	-

mon.q	y	9
	PC	3
	return	-

ASOCIACIÓN DINÁMICA DE MÉTODOS - EJECUCIÓN PASO A PASO

```
      class A {
          int a = 4, b = 9

          fun p(int x): int {
▸ 0           a = b * x
▸ 1           return q(a)
          }

          fun q(int y): int {
▸ 2           return a + y
          }
      }

      class B extends A {
          A sopa = new C()

          fun q(int y): int {
▸ 3           return sopa.p(a + b) + y
          }
      }

      class C extends B {
          int c = 0

          fun p(int x): int {
▸ 4           a = 3 + x
▸ 5           c = 2 * b − x
▸ 6           return q(a + b + c)
          }

          fun q(int y): int {
▸ 7           return c + y
          }
      }

▸ 8  A mon = new B()
▸ 9  A don = new C()
▸ 10 B go = new C()

▸ 11 print(mon.p(1) + don.p(1) + go.p(1))
```

Nombre	TE	TD	Methods table	Values
mon	A	B	p -> A, q -> B	a = 9, b = 9
don	A	C	p -> C, q -> C	a = 4, b = 9, c = 0
go	B	C	p -> C, q -> C	a = 4, b = 9, c = 0
mon.sopa	A	C	p -> C, q -> C	a = 21, b = 9, c = 0
don.sopa	A	C	p -> C, q -> C	a = 4, b = 9, c = 0
go.sopa	A	C	p -> C, q -> C	a = 4, b = 9, c = 0

mon.sopa.p	x	18
	PC	6
	return	30

mon.q	y	9
	PC	3
	return	-

mon.p	x	1
	PC	1
	return	-

ASOCIACIÓN DINÁMICA DE MÉTODOS - EJECUCIÓN PASO A PASO

```
     class A {
         int a = 4, b = 9

         fun p(int x): int {
▸ 0          a = b * x
▸ 1          return q(a)
         }

         fun q(int y): int {
▸ 2          return a + y
         }
     }

     class B extends A {
         A sopa = new C()

         fun q(int y): int {
▸ 3          return sopa.p(a + b) + y
         }
     }

     class C extends B {
         int c = 0

         fun p(int x): int {
▸ 4          a = 3 + x
▸ 5          c = 2 * b − x
▸ 6          return q(a + b + c)
         }

         fun q(int y): int {
▸ 7          return c + y
         }
     }

▸ 8  A mon = new B()
▸ 9  A don = new C()
▸ 10 B go = new C()

▸ 11 print(mon.p(1) + don.p(1) + go.p(1))
```

Nombre	TE	TD	Methods table	Values
mon	A	B	p -> A, q -> B	a = 9, b = 9
don	A	C	p -> C, q -> C	a = 4, b = 9, c = 0
go	B	C	p -> C, q -> C	a = 4, b = 9, c = 0
mon.sopa	A	C	p -> C, q -> C	a = 21, b = 9, c = 0
don.sopa	A	C	p -> C, q -> C	a = 4, b = 9, c = 0
go.sopa	A	C	p -> C, q -> C	a = 4, b = 9, c = 0

mon.q	y	9
	PC	3
	return	30 + 9 = 39

mon.p	x	1
	PC	1
	return	-

Global	PC	11

ASOCIACIÓN DINÁMICA DE MÉTODOS - EJECUCIÓN PASO A PASO

```
        class A {
            int a = 4, b = 9

            fun p(int x): int {
▸ 0             a = b * x
▸ 1             return q(a)
            }

            fun q(int y): int {
▸ 2             return a + y
            }
        }

        class B extends A {
            A sopa = new C()

            fun q(int y): int {
▸ 3             return sopa.p(a + b) + y
            }
        }

        class C extends B {
            int c = 0

            fun p(int x): int {
▸ 4             a = 3 + x
▸ 5             c = 2 * b - x
▸ 6             return q(a + b + c)
            }

            fun q(int y): int {
▸ 7             return c + y
            }
        }

▸ 8   A mon = new B()
▸ 9   A don = new C()
▸ 10  B go = new C()

▸ 11  print(mon.p(1) + don.p(1) + go.p(1))
```

Nombre	TE	TD	Methods table	Values
mon	A	B	p -> A, q -> B	a = 9, b = 9
don	A	C	p -> C, q -> C	a = 4, b = 9, c = 0
go	B	C	p -> C, q -> C	a = 4, b = 9, c = 0
mon.sopa	A	C	p -> C, q -> C	a = 21, b = 9, c = 0
don.sopa	A	C	p -> C, q -> C	a = 4, b = 9, c = 0
go.sopa	A	C	p -> C, q -> C	a = 4, b = 9, c = 0

mon.p	x	1
	PC	1
	return	39

Global	PC	11

ASOCIACIÓN DINÁMICA DE MÉTODOS - EJECUCIÓN PASO A PASO

```
      class A {
          int a = 4, b = 9

          fun p(int x): int {
▸  0          a = b * x
▸  1          return q(a)
          }

          fun q(int y): int {
▸  2          return a + y
          }
      }

      class B extends A {
          A sopa = new C()

          fun q(int y): int {
▸  3          return sopa.p(a + b) + y
          }
      }

      class C extends B {
          int c = 0

          fun p(int x): int {
▸  4          a = 3 + x
▸  5          c = 2 * b − x
▸  6          return q(a + b + c)
          }

          fun q(int y): int {
▸  7          return c + y
          }
      }

▸  8  A mon = new B()
▸  9  A don = new C()
▸ 10  B go = new C()

▸ 11  print(~~mon.p(1)~~ 39 + don.p(1) + go.p(1))
```

Nombre	TE	TD	Methods table	Values
mon	A	B	p -> A, q -> B	a = 9, b = 9
don	A	C	p -> C, q -> C	a = 4, b = 9, c = 0
go	B	C	p -> C, q -> C	a = 4, b = 9, c = 0
mon.sopa	A	C	p -> C, q -> C	a = 21, b = 9, c = 0
don.sopa	A	C	p -> C, q -> C	a = 4, b = 9, c = 0
go.sopa	A	C	p -> C, q -> C	a = 4, b = 9, c = 0

Global	PC	11

ASOCIACIÓN DINÁMICA DE MÉTODOS - EJECUCIÓN PASO A PASO

```
        class A {
            int a = 4, b = 9

            fun p(int x): int {
▸  0            a = b * x
▸  1            return q(a)
            }

            fun q(int y): int {
▸  2            return a + y
            }
        }

        class B extends A {
            A sopa = new C()

            fun q(int y): int {
▸  3            return sopa.p(a + b) + y
            }
        }

        class C extends B {
            int c = 0

            fun p(int x): int {
▸  4            a = 3 + x
▸  5            c = 2 * b - x
▸  6            return q(a + b + c)
            }

            fun q(int y): int {
▸  7            return c + y
            }
        }

▸  8    A mon = new B()
▸  9    A don = new C()
▸ 10    B go = new C()

▸ 11    print(~~mon.p(1)~~ 39 + don.p(1) + go.p(1))
```

Nombre	TE	TD	Methods table	Values
mon	A	B	p -> A, q -> B	a = 9, b = 9
don	A	C	p -> C, q -> C	a = 3 + 1, b = 9, c = 18 - 1
go	B	C	p -> C, q -> C	a = 4, b = 9, c = 0
mon.sopa	A	C	p -> C, q -> C	a = 21, b = 9, c = 0
don.sopa	A	C	p -> C, q -> C	a = 4, b = 9, c = 0
go.sopa	A	C	p -> C, q -> C	a = 4, b = 9, c = 0

don.p	x	1
	PC	6
	return	-

Global	PC	11

ASOCIACIÓN DINÁMICA DE MÉTODOS - EJECUCIÓN PASO A PASO

```
       class A {
           int a = 4, b = 9

           fun p(int x): int {
 0             a = b * x
 1             return q(a)
           }

           fun q(int y): int {
 2             return a + y
           }
       }

       class B extends A {
           A sopa = new C()

           fun q(int y): int {
 3             return sopa.p(a + b) + y
           }
       }

       class C extends B {
           int c = 0

           fun p(int x): int {
 4             a = 3 + x
 5             c = 2 * b - x
 6             return q(a + b + c)
           }

           fun q(int y): int {
 7             return c + y
           }
       }

 8     A mon = new B()
 9     A don = new C()
10     B go = new C()

11     print(~~mon.p(1)~~ 39 + don.p(1) + go.p(1))
```

Nombre	TE	TD	Methods table	Values
mon	A	B	p -> A, q -> B	a = 9, b = 9
don	A	C	p -> C, q -> C	a = 4, b = 9, c = 17
go	B	C	p -> C, q -> C	a = 4, b = 9, c = 0
mon.sopa	A	C	p -> C, q -> C	a = 21, b = 9, c = 0
don.sopa	A	C	p -> C, q -> C	a = 4, b = 9, c = 0
go.sopa	A	C	p -> C, q -> C	a = 4, b = 9, c = 0

don.p	x	1
	PC	6
	return	-

Global	PC	11

ASOCIACIÓN DINÁMICA DE MÉTODOS - EJECUCIÓN PASO A PASO

```
     class A {
         int a = 4, b = 9

         fun p(int x): int {
 0           a = b * x
 1           return q(a)
         }

         fun q(int y): int {
 2           return a + y
         }
     }

     class B extends A {
         A sopa = new C()

         fun q(int y): int {
 3           return sopa.p(a + b) + y
         }
     }

     class C extends B {
         int c = 0

         fun p(int x): int {
 4           a = 3 + x
 5           c = 2 * b - x
 6           return q(a + b + c)
         }

         fun q(int y): int {
 7           return c + y
         }
     }

 8   A mon = new B()
 9   A don = new C()
10   B go = new C()

11   print(~~mon.p(1)~~ 39 + don.p(1) + go.p(1))
```

Nombre	TE	TD	Methods table	Values
mon	A	B	p -> A, q -> B	a = 9, b = 9
don	A	C	p -> C, q -> C	a = 4, b = 9, c = 17
go	B	C	p -> C, q -> C	a = 4, b = 9, c = 0
mon.sopa	A	C	p -> C, q -> C	a = 21, b = 9, c = 0
don.sopa	A	C	p -> C, q -> C	a = 4, b = 9, c = 0
go.sopa	A	C	p -> C, q -> C	a = 4, b = 9, c = 0

don.q	y	30
	PC	7
	return	17 + 30 = 47

don.p	x	1
	PC	6
	return	-

Global	PC	11

ASOCIACIÓN DINÁMICA DE MÉTODOS - EJECUCIÓN PASO A PASO

```
      class A {
          int a = 4, b = 9

          fun p(int x): int {
▸ 0           a = b * x
▸ 1           return q(a)
          }

          fun q(int y): int {
▸ 2           return a + y
          }
      }

      class B extends A {
          A sopa = new C()

          fun q(int y): int {
▸ 3           return sopa.p(a + b) + y
          }
      }

      class C extends B {
          int c = 0

          fun p(int x): int {
▸ 4           a = 3 + x
▸ 5           c = 2 * b - x
▸ 6           return q(a + b + c)
          }

          fun q(int y): int {
▸ 7           return c + y
          }
      }

▸ 8  A mon = new B()
▸ 9  A don = new C()
▸ 10 B go = new C()

▸ 11 print(~~mon.p(1)~~ 39 + don.p(1) + go.p(1))
```

Nombre	TE	TD	Methods table	Values
mon	A	B	p -> A, q -> B	a = 9, b = 9
don	A	C	p -> C, q -> C	a = 4, b = 9, c = 17
go	B	C	p -> C, q -> C	a = 4, b = 9, c = 0
mon.sopa	A	C	p -> C, q -> C	a = 21, b = 9, c = 0
don.sopa	A	C	p -> C, q -> C	a = 4, b = 9, c = 0
go.sopa	A	C	p -> C, q -> C	a = 4, b = 9, c = 0

don.p	x	1
	PC	6
	return	47

Global	PC	11

ASOCIACIÓN DINÁMICA DE MÉTODOS - EJECUCIÓN PASO A PASO

```
      class A {
          int a = 4, b = 9

          fun p(int x): int {
 0            a = b * x
 1            return q(a)
          }

          fun q(int y): int {
 2            return a + y
          }
      }

      class B extends A {
          A sopa = new C()

          fun q(int y): int {
 3            return sopa.p(a + b) + y
          }
      }

      class C extends B {
          int c = 0

          fun p(int x): int {
 4            a = 3 + x
 5            c = 2 * b − x
 6            return q(a + b + c)
          }

          fun q(int y): int {
 7            return c + y
          }
      }

 8    A mon = new B()
 9    A don = new C()
10    B go = new C()

11    print(~~mon.p(1)~~ 39 + ~~don.p(1)~~ 47 + go.p(1))
```

Nombre	TE	TD	Methods table	Values
mon	A	B	p -> A, q -> B	a = 9, b = 9
don	A	C	p -> C, q -> C	a = 4, b = 9, c = 17
go	B	C	p -> C, q -> C	a = 4, b = 9, c = 0
mon.sopa	A	C	p -> C, q -> C	a = 21, b = 9, c = 0
don.sopa	A	C	p -> C, q -> C	a = 4, b = 9, c = 0
go.sopa	A	C	p -> C, q -> C	a = 4, b = 9, c = 0

Global	PC	11

ASOCIACIÓN DINÁMICA DE MÉTODOS - EJECUCIÓN PASO A PASO

```
       class A {
           int a = 4, b = 9

           fun p(int x): int {
‣ 0            a = b * x
‣ 1            return q(a)
           }

           fun q(int y): int {
‣ 2            return a + y
           }
       }

       class B extends A {
           A sopa = new C()

           fun q(int y): int {
‣ 3            return sopa.p(a + b) + y
           }
       }

       class C extends B {
           int c = 0

           fun p(int x): int {
‣ 4            a = 3 + x
‣ 5            c = 2 * b - x
‣ 6            return q(a + b + c)
           }

           fun q(int y): int {
‣ 7            return c + y
           }
       }

‣ 8  A mon = new B()
‣ 9  A don = new C()
‣ 10 B go = new C()

‣ 11 print(~~mon.p(1)~~ 39 + ~~don.p(1)~~ 47 + go.p(1))
```

Nombre	TE	TD	Methods table	Values
mon	A	B	p -> A, q -> B	a = 9, b = 9
don	A	C	p -> C, q -> C	a = 4, b = 9, c = 17
go	B	C	p -> C, q -> C	a = 4, b = 9, c = 0
mon.sopa	A	C	p -> C, q -> C	a = 21, b = 9, c = 0
don.sopa	A	C	p -> C, q -> C	a = 4, b = 9, c = 0
go.sopa	A	C	p -> C, q -> C	a = 4, b = 9, c = 0

El objeto go tiene el mismo *tipo dinámico* y los *mismos valores* *<u>iniciales</u>* para a, b y c que el objeto don.

Por lo tanto los resultados (y pasos) de ejecución de go.p(1) son idénticos a los de don.p(1).

Se omiten los pasos, en la siguiente lamina están los resultados.

Global	PC	11

ASOCIACIÓN DINÁMICA DE MÉTODOS - EJECUCIÓN PASO A PASO

```
     class A {
         int a = 4, b = 9

         fun p(int x): int {
 0           a = b * x
 1           return q(a)
         }

         fun q(int y): int {
 2           return a + y
         }
     }

     class B extends A {
         A sopa = new C()

         fun q(int y): int {
 3           return sopa.p(a + b) + y
         }
     }

     class C extends B {
         int c = 0

         fun p(int x): int {
 4           a = 3 + x
 5           c = 2 * b - x
 6           return q(a + b + c)
         }

         fun q(int y): int {
 7           return c + y
         }
     }

 8   A mon = new B()
 9   A don = new C()
10   B go = new C()

11   print(~~mon.p(1)~~ 39 + ~~don.p(1)~~ 47 + ~~go.p(1)~~ 47)
```

Nombre	TE	TD	Methods table	Values
mon	A	B	p -> A, q -> B	a = 9, b = 9
don	A	C	p -> C, q -> C	a = 4, b = 9, c = 17
go	B	C	p -> C, q -> C	a = 4, b = 9, c = 17
mon.sopa	A	C	p -> C, q -> C	a = 21, b = 9, c = 0
don.sopa	A	C	p -> C, q -> C	a = 4, b = 9, c = 0
go.sopa	A	C	p -> C, q -> C	a = 4, b = 9, c = 0

IMPRIME

```
> 133
```

Global	PC	11

ASOCIACIÓN DINÁMICA DE MÉTODOS

RESULTADOS

FINALMENTE

El programa imprime:

```
A mon = new B()
A don = new C()
B go = new C()

print(mon.p(1) + don.p(1) + go.p(1))



> 133
```