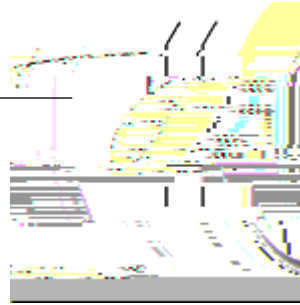




ming languages.

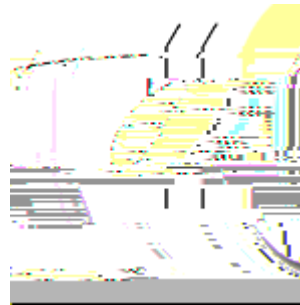
Agents are a new type of objects that allow to store code to be executed and

```
print_coordinates(x, y: INTEGER) is
  do
    i.o.put_integer(x) ; i.o.put_character(' ')
    i.o.put_integer(y) ; i.o.put_new_line
  end
```



do

...



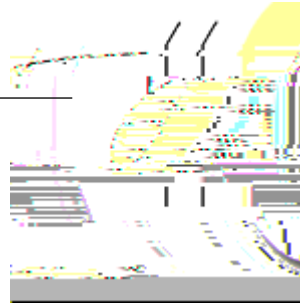
The following example shows the basic use of `do_all`:

```
zoo: ARRAY[ANIMAL]
```

```
foo is
do
    ...
    zoo.do_all ( agent print_name(?) )      -- (5)
    ...
end
```

```
print_name(item: ANIMAL) is
do
    i.o.put_string( item.name ) ; i.o.put_new_line  -- (6)
end
```

agent









```
...
end
class DOG -- Extract
  eat(meat: MEAT) is -- (10)
  do
    ...
  end
```

If we consider line (8)

### 3 STANDARD EIFFEL CONFORMANCE RULES











they rely on the code for the *WINDOW* class provided on page 127:

```
my_window.when_pointer_move( agent print_all )           -- (27)
my_window.when_pointer_move( agent print_x )           -- (28)
my_window.when_pointer_move( agent iO.90.put_string("move%N") ) -- (29)
```

```
print_x(x: INTEGER) is
do
    iO.90.put_integer(x) ; iO.put_new_line
end
```

```
print_all(x, y:
```



