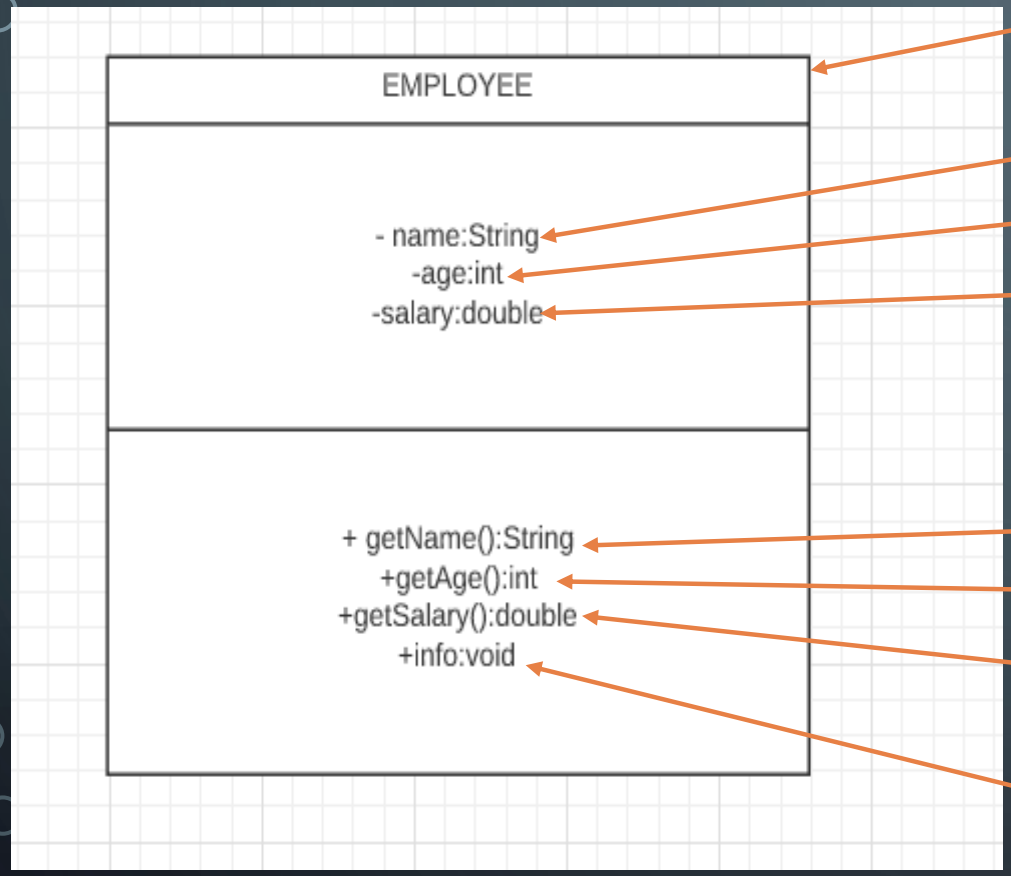
ΑΝΤΚΕΙΜΕΝΟΣΤΡΑΦΗΣ UML

UML-ΣΥΜΒΟΛΑ

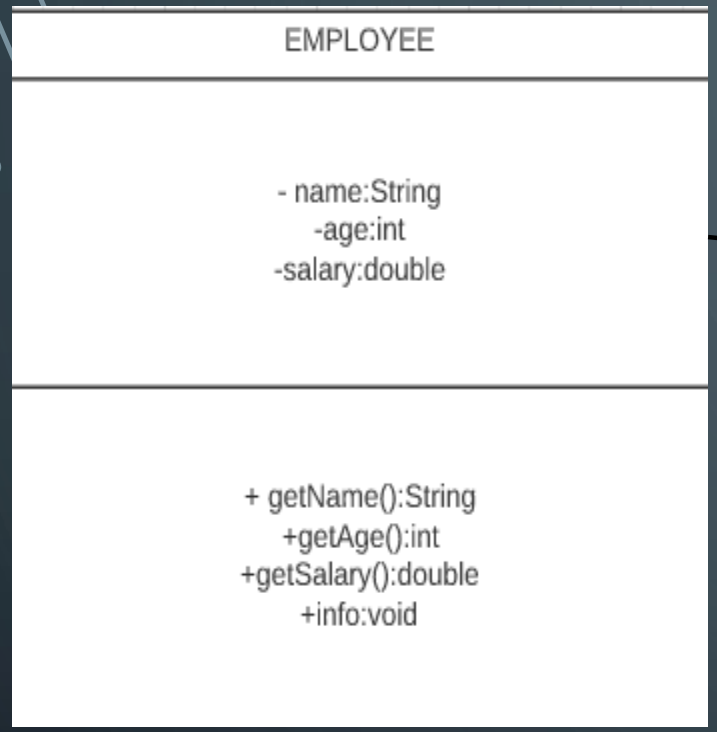


Όνομα κλάσης EMPLOYEE.

Ιδιωτική μεταβλητή(-) με όνομα name και τύπο δεδομένων String.
Ιδιωτική μεταβλητή(-) με το όνομα age και τύπο int.
Ιδιωτική μεταβλητή(-) με το όνομα salary και τύπο double.

Δημόσια συνάρτηση μέλος(+) με το όνομα getName και τύπο επιστροφής String.
Δημόσια συνάρτηση μέλος(+) με το όνομα getAge και τύπο επιστροφής int.
Δημόσια συνάρτηση μέλος(+) με το όνομα getSalary και τύπο επιστροφής double.
Δημόσια συνάρτηση μέλος(+) με το όνομα getName και χωρίς τύπο επιστροφής.

UML-ΜΕΤΑΤΡΟΠΗ ΣΕ ΚΩΔΙΚΑ




```
class employee
{
    std::string name;
    int age;
    double salary;
    public:
      employee(std::string n,int a,double s):name(n),age(a),salary(s)
      {}
      std::string getName()const {return this->name;}
      int getAge()const {return this->age;}
      double getSalary()const {return this->salary;}
      void info()
      {
          std::cout<<this->name<<"-"<<this->age<<"-"
           <<this->salary<<std::endl;
      }
      friend std::ostream &operator<<(std::ostream &os,const employee &e)
      {
          return os<<e.name<<"-"<<e.age<<"-"<<e.salary<<std::endl;
      }
};
```

UML-ΣΥΜΒΟΛΙΣΜΟΙ

#→Protected πεδίο η συνάρτηση μέλος.
-→Private πεδίο η συνάρτηση μέλος.
+→Public πεδίο η συνάρτηση μέλος.

1ο Επίπεδο→Δηλώνεται το όνομα της κλάσης
2ο Επιπέδο→Δηλώνονται τα πεδία της κλάσης και το επίπεδο προσβασιμότητας του.
3ο Επίπεδο→Δηλώνονται η συναρτήσεις μέλη της κλάσης και το επίπεδο προσβασιμότητας του.

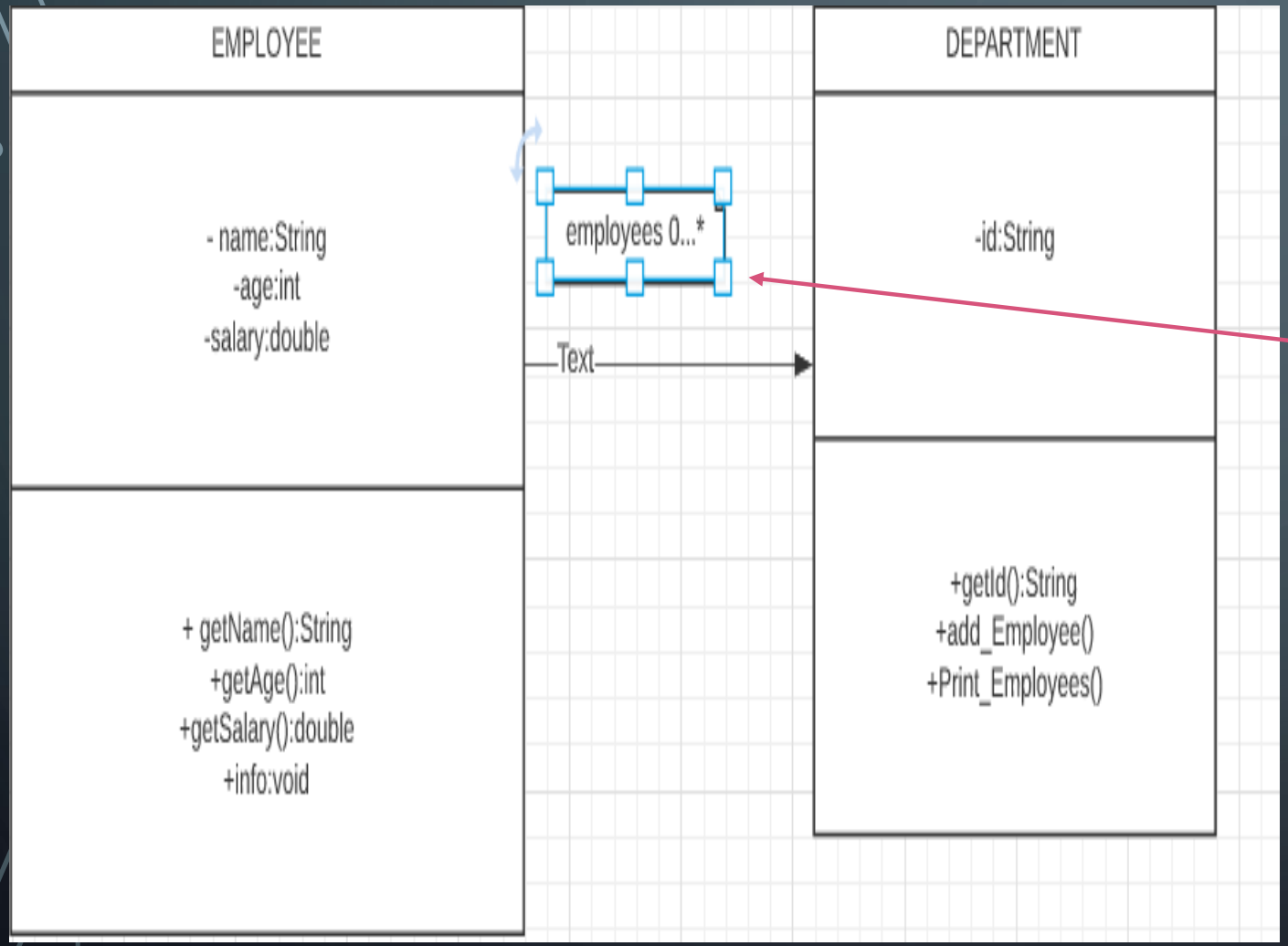
ΣΥΝΤΑΞΗ

→Πρώτα δηλώνεται το όναμα της μεταβλητής συνάρτησης/πεδίου,μετά →: και έπειτα ο τύπος επιστροφής της συνάρτησης /πεδίου.

→Παράδειγμα

- +getName():String
- -name:String
- #id:int

UML-ΣΥΜΒΟΛΙΣΜΟΙ



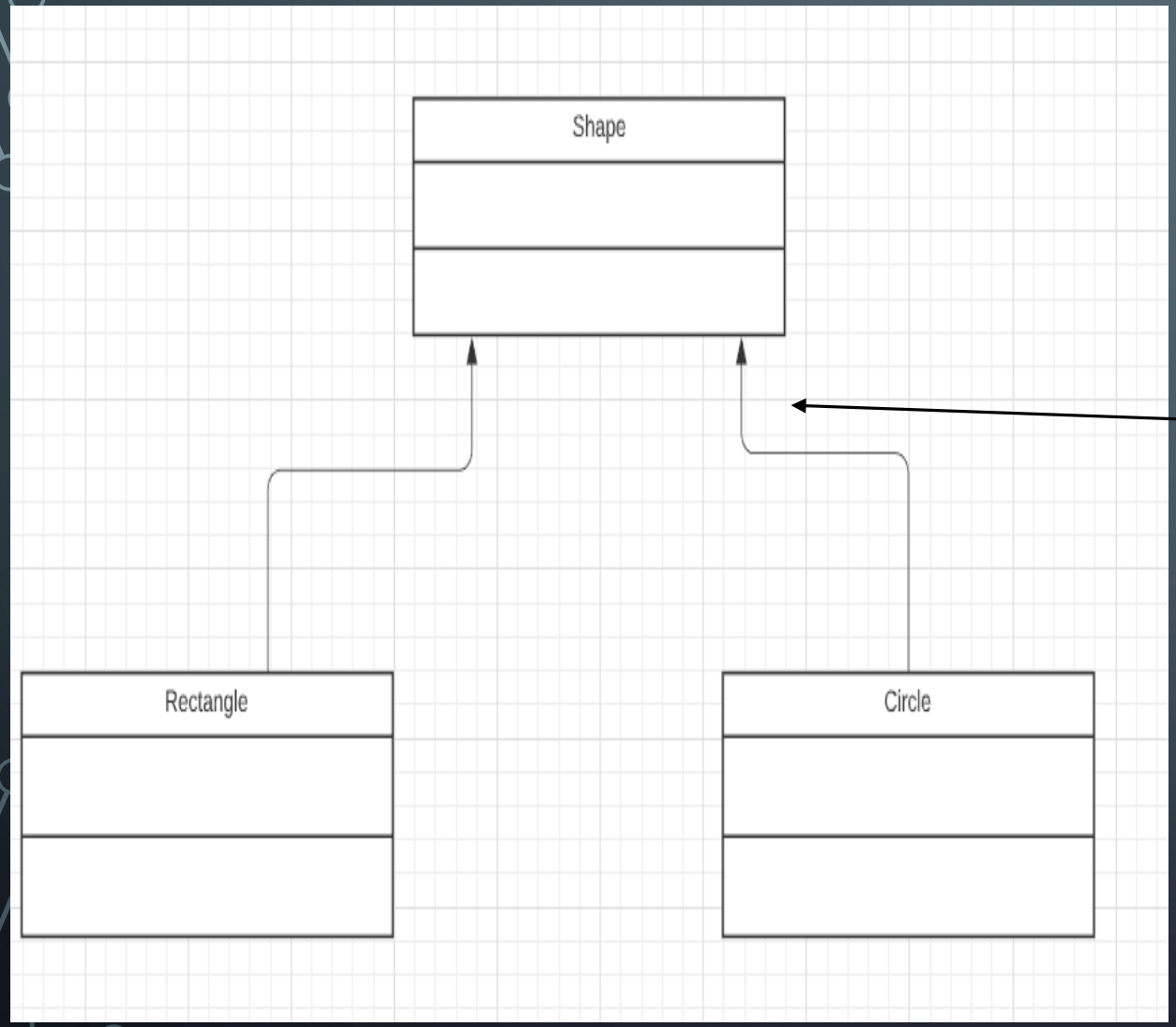
Δήλωση συσχέτησης μεταξύ της κλάσης Employee και της κλάσης Department.Στην κλάση Department θα υπάρχει ένα container σαν ιδιωτικό πεδίο που θα αποθηκεύει αντικείμενα τύπου employee.Εξάρτηση,αν η κλάση Department εξαρτώταν από την δημιουργία αντικειμένου Employee.

UML-ΥΛΟΠΟΙΗΣΗ ΣΕ ΚΩΔΙΚΑ

```
class employee
{
    std::string name;
    int age;
    double salary;
    public:
      employee(std::string n,int a,double s):name(n),age(a),salary(s)
      {}
      std::string getName()const {return this->name;}
      int getAge()const {return this->age;}
      double getSalary()const {return this->salary;}
      void info()
      {
          return this->name+"-"+std::to_string(this->age)+"-"
          +std::to_string(this->salary);
      }
      friend std::ostream &operator<<(std:::ostream &os,const employee &e)
      {
          return os<<e.name<<"-"<<e.age<<"-"<<e.salary<<std::endl;
      }
};
```

```
class department
{
    private:
      int id;
      std::vector <employee> emps;
    public:
      department(int i):id(i) {}
      void add_employee(const employee &e)
      {
          this->emps.push_back(e);
      }
      int getId()const {return this->id;}
      void Print_Employees()
      {
         std::ostream_iterator <employee> out(std::cout,"\n");
         std::copy(this->emps.begin(),this->emps.end(),out);
      }
};
```

UML-ΣΥΜΒΟΛΙΣΜΟΙ



Δήλωση κληρονομικότητας.Οι κλάσεις Rectangle και Circle κληρονομούν την κλάση βάση shape.

UML-ΥΛΟΠΟΙΗΣΗ ΣΕ ΚΩΔΙΚΑ

```
#include <iostream>

class shape
{
    //members and operations
};

class Rectangle:public shape
{
    //members and operations
};

class circle:public shape{
    //members and operations
};
```

ΚΩΔΙΚΕΣ UML

- EMPLOYEE C++
- DEPARTMENT C++
- INHERITANCE C++
- EMPLOYEE JAVA
- DEPARTMENT JAVA
- INHERITANCE JAVA