

Explicație - Constructor

Petculescu Mihai-Silviu

Explicație - Constructor

Petculescu Mihai-Silviu

1. Se cere să explicați rezultatele obținute prin executarea programului C++ de mai jos.
2. Corectați programul, astfel ca pe cele doua linii să fie afișată aceeași valoare(20).
3. Rescrieți în Java ambele variante.

1. Se cere să explicați rezultatele obținute prin executarea programului C++ de mai jos.

Rezultat

```
20
0
```

Deoarece în programul dat, în clasa Person, vârsta este salvată doar ca și o variabilă locală `age` programul va afișa vârsta trimisă ca și parametru, în cazul nostru `20`, și prin intermediul liniei `cout<<p.age<<endl` din `main()` valoarea albitrară alocată de compiler la construcția acelei instanțe a clasei (care poate fi 0 sau orice altă valoare, depinzând de compilerul c++ folosit și, eventual, de momentul execuției).

2. Corectați programul, astfel ca pe cele doua linii să fie afișată aceeași valoare(20).

Prin modificarea liniei `int age=a;` în `age=a;` (sau mai precis prin eliminarea re-declarării variabilei `age` în interiorul constructorului clasei `Person`), valoarea variabilei va fi memorată în zona de memorie asociată instanței clasei, și nu într-un sector random de memorie, ca în exemplul precedent.

3. Rescrieți în Java ambele variante.

Implementare prima variantă

Person.java

```
public class Person {
    int age;

    public Person(int a){
        age=a;
        System.out.println(age);
    }
}
```

Main.java

```
public class Main {  
    public static void main(String args[]){  
        Person p = new Person(20);  
        System.out.println(p.age);  
    }  
}
```

Implementare a doua variantă

Person.java

```
public class Person {  
    int age;  
  
    public Person(int a){  
        int age=a;  
        System.out.println(age);  
    }  
}
```

Main.java

```
public class Main {  
    public static void main(String args[]){  
        Person p = new Person(20);  
        System.out.println(p.age);  
    }  
}
```