Test POO - versiunea C

C	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
a																				
b																				
c																				
d																				

1. Când apare `NullPointerException`?

   (a) Când încercam sa deschidem un fișier care nu există
   (b) Când facem buffer overflow la un array
   (c) Când încercam să folosim un obiect null
   (d) Când folosim o colecție care este instanțiata și goală

2. Care este rezultatul execuției codului de mai jos?

```
String a = "hello";
String b = "world";
String c = b;
b = a + b;
String d = new String("hello");
System.out.println(c + " " + (d == a));
```

   (a) world false
   (b) helloworld true
   (c) hello true
   (d) helloworld false

3. Care variantă definește cel mai bine legătura dintre interfețe și clase?

   (a) Nu pot exista relații între clase și interfețe
   (b) Nici clasele, nici interfețele nu precizează modul în care un obiect execută o operație
   (c) Interfețele precizeaza operațiile expuse de un obiect, în timp ce clasele modul în care acesta le execută
   (d) Atât clasele, cat și interfețele definesc modul în care un obiect execută o operație

4. Ce afirmație despre Set si List este adevarată?

   (a) In Set nu avem elemente duplicate, iar Set extinde interfața List
   (b) In Set putem avea elemente duplicate, la fel ca în List
   (c) Nici in Set, nici în List nu avem elemente duplicate
   (d) In Set nu avem elemente duplicate, iar în List putem să avem

5. Fie:

```
class Device {
        public static void charge() {
                System.out.print("Charging device; ");
        }
}
class Phone extends Device {
        public static void charge() {
                System.out.print("Charging phone; ");
        }
}
```

   Care este rezultatul execuției codului de mai jos?

```
Device phone = new Phone();
Device device = new Device();
phone.charge();
device.charge();
Device.charge();
```

   (a) Charging device; Charging device; Charging phone;
   (b) Charging phone; Charging phone; Charging device;
   (c) Charging device; Charging device; Charging device;
   (d) Charging phone; Charging device; Charging device;

6. Care afirmații sunt adevarate despre clase interne statice?
   1. Nu au acces la membri non-statici ai clasei exterioare
   2. Este nevoie de o instanță a clasei externe pentru a o instanția
   3. Trebuie sa moștenească clasa exterioară
   4. Trebuie instanțiată astfel: Outer out = new Outer(); Inner in = out.new Inner();

   (a) 3
   (b) 1
   (c) 2,4
   (d) 1,2

7. In Java, prin compararea a doua obiecte cu ==, ce se compara?

   (a) tipurile obiectelor
   (b) marimile obiectelor
   (c) adresele obiectelor
   (d) valorile obiectelor

8. Fie un serviciu de tip Youtube, în care avem utilizatori care se pot abona la canale, putând fi la curent cu noutățile canalelor la care ei sunt abonați (încarcare videoclip nou, depășirea unui anumit numar de abonați etc). Ce design pattern se poate aplica aici?

   (a) Visitor
   (b) Factory
   (c) Observer
   (d) Command

9. Care dintre următoarele afirmații sunt adevarate?
   1) Clasele abstracte pot avea keyword-ul final
   2) O clasă poate moșteni mai multe clase în același timp
   3) O interfață poate moșteni mai multe interfețe în același timp
   4) Metodele și atributele de tip protected pot fi accesate doar din clasa-origine și din clasele derivate

   (a) 1, 2, 3
   (b) 3, 4
   (c) 2, 3, 4
   (d) 1, 2, 3, 4

10. Dacă `Feline` extinde clasa abstractă `Animal` și `Lion` extinde `Feline`, atunci care instanțiere este corectă?

    (a) Feline feline = new Animal();
    (b) Animal animal = new Feline();
    (c) Lion lion = new Animal();
    (d) Lion lion = new Feline();

11. Ce înseamnă constructorul *default*?

    (a) constructor fără parametri declarat de utilizator
    (b) constructor fără implementare
    (c) constructor fără modificatori de acces
    (d) constructor fără parametri adăugat de Java dacă nici un constructor nu a fost declarat

12. Ce design pattern putem folosi pentru a avea un obiect unic și vizibil la nivel global?

    (a) Observer
    (b) Visitor
    (c) Singleton
    (d) Factory

13. In ce colectie de tip Set se pastreaza ordinea insertiei elementelor?

    (a) AbstractSet
    (b) LinkedHashSet
    (c) HashSet
    (d) TreeSet

14. Ce colecție ar fi mai eficientă de folosit dacă dorim să stocăm o secvență de elemente pe care să o modificăm rar dar pe care să o accesăm foarte des?

    (a) ArrayList
    (b) LinkedList
    (c) niciuna din variante
    (d) Vector

15. Ce rol are un *copy constructor*?

(a) de a copia datele și referința unui obiect
(b) de a copia referința unui obiect
(c) de a distruge referința unui obiect
(d) de a copia datele unui obiect

16. Care este rezultatul execuției codului de mai jos?

```
Food food1 = new Food();
Food food2 = new Cake();
Cake cake = new Cake();
food1.cut();
food2.cut();
cake.cut();
```

Unde:

```
class Cake extends Food {
        public void cut() {
                System.out.print(" Cake ");
        }
}
class Food {
        public void cut() {
                System.out.print(" Food ");
        }
}
```

(a) Food Food Cake
(b) Food Food Food
(c) Food Cake Cake
(d) Cake Cake Cake

17. Identificați pe care linii avem suprascriere (overriding) și pe care avem supraîncărcare (overloading)

```
interface Coffee {
    void brew();   // 1
    void drink();  // 2
}
class Espresso implements Coffee {
    public void brew() { }       // 3
    public void drink() { }      // 4
    public void drink(Boolean withSugar) { } // 5
    public void drink(Boolean withMilk,
                      Boolean withSugar) { } // 6
}
```

(a) Suprascriere: 5,6; Supraîncărcare: 3,4
(b) Suprascriere: 1,2,3; Supraîncărcare: 4,5,6
(c) Suprascriere: 3; Supraîncărcare: 4,5,6
(d) Suprascriere: 3,4; Supraîncărcare: 5,6

18. Ce se va afisa la rularea codului?

```
class Foo {
    public int doWork(int x) {
        try {
            x++;
            if (x < 3) {
                throw new Exception();
            }
            return ++x;
        } catch (Exception e) {
            return x++;
        } finally {
            return ++x;
        }
    }
}
class Main {
    public static void main(String[] args) {
        Foo f = new Foo();
        System.out.println(f.doWork(1));
    }
}
```

(a) 3
(b) 4
(c) 2
(d) 5

19. Ce se va afisa la rularea codului?

```
class Foo {
    public static void doWork() {
        System.out.print(Done! ");
    }
}
class Main {
    public static void main(String[] args) {
        Foo f = null;
        f.doWork();
        Foo.doWork();
    }
}
```

(a) "Done! Done!"
(b) La compilare va da o eroare la f.doWork();
(c) "Done! "
(d) La rulare programul va da NullPointerException

20. Ce reprezinta implementarea unei clase anonime?

(a) implementarea unei interfețe sau extinderea unei clase
(b) implementarea de metode anonime
(c) implementarea unei interfețe în mod exclusiv
(d) crearea unei clase care are un constructor