

Examen 1_ ANA LAURA VÁZQUEZ SOLACHE

1. Paradigmas de programación en Java

- a. POO, reactiva, funcional.

2. Cuales son los componentes de la programación orientada a objetos y describirlos

- a. Variables de referencia, referencias y objetos.

Por ejemplo si generamos una clase Pato, podríamos instanciar al Pato de la siguiente manera

```
Pato Donald = new Pato;           //en este ejemplo, la variable de referencia Pato, con  
                                   referencia Dondald, está apuntando al objeto new Pato.
```

3. Diferencias entre un String y un StringBuilder

- a. El .equals() se comporta diferente para cada uno, lo que pasa en el StringBuilder, que al heredar de Object, mantiene su uso siendo este dar como resultado del método la dirección del objeto en memoria; y de el String está sobrescrito y lo que hace es comparar el contenido.
- b. El objeto del String es inmutable y el objeto del StringBuilder es mutable.

4. Explica que es un null

- a. Cuando una variable de referencia no apunta a ningún objeto(no está inicializada).

5. ¿Cuál es el constructor default?

- a. El constructor por defecto se crea por el IDE con atributos del objeto con valores predeterminados, sin parámetros. Estos se crean con modificador de acceso "public" y una vez que nosotros ya definimos constructores, este deja de existir, por lo que si queremos utilizarlo (un constructor sin parámetros) hay que escribir su código.

Examen 1_ ANA LAURA VÁZQUEZ SOLACHE

7. Menciona los tipos de variables y sus propiedades

- a. Los tipos de variables son: instancia de clase, clase y variable local
Una variable de instancia de clase, es cuando estas están definidas dentro de una clase.
Las variables de clase son aquellas que cuentan con la palabra reservada "static"
Y finalmente una variable local, es cuando estas se encuentran dentro de algún método.

2. What lines are printed by the following program? (Choose all that apply.)

```
1: public class WaterBottle {  
2:     private String brand;  
3:     private boolean empty;  
4:     public static float code;  
5:     public static void main(String[] args) {  
6:         WaterBottle wb = new WaterBottle();  
7:         System.out.println("Empty = " + wb.empty);  
8:         System.out.println("Brand = " + wb.brand);  
9:         System.out.println("Code = " + code);  
10:    } }
```

- A. Line 8 generates a compiler error.
B. Line 9 generates a compiler error.
C. Empty =
D. Empty = false
E. Brand =
F. Brand = null
G. Code = 0.0
H. Code = 0f

Las respuestas correctas son **D, F y G** lo que pasa aquí es que las variables al no estar inicializadas muestran su valor por default(defecto) al ser ejecutadas.

Todos los atributos cuentan con modificadores de acceso "private" y "public" los cuales pueden ser usados dentro de la misma clase por lo cual no hay problema ni necesidad de importaciones o herencias.

Dentro de nuestro método main, que es el punto de entrada de nuestro programa, estamos instanciando la clase "WaterBottle" e imprimiendo los valores por defecto de sus atributos al no estar inicializados.

10. Which statements about the following class are correct? (Choose all that apply.)

```
1: public class PoliceBox {  
2:     String color;  
3:     long age;  
4:     public void PoliceBox() {  
5:         color = "blue";  
6:         age = 1200;  
7:     }  
8:     public static void main(String []time) {  
9:         var p = new PoliceBox();  
10:        var q = new PoliceBox();  
11:        p.color = "green";  
12:        p.age = 1400;  
13:        p = q;  
14:        System.out.println("Q1="+q.color);  
15:        System.out.println("Q2="+q.age);  
16:        System.out.println("P1="+p.color);  
17:        System.out.println("P2="+p.age);  
18: } }
```

- A. It prints Q1=blue.
- B. It prints Q2=1200.
- C. It prints P1=null.**
- D. It prints P2=1400.
- E. Line 4 does not compile.
- F. Line 12 does not compile.
- G. Line 13 does not compile.
- H. None of the above

La respuesta correcta es C.

Aquí estamos generando una clase "PoliceBox" la cual cuenta con los atributos color & age con el método PoliceBox(no es un constructor porque tiene el retorno de "void")

En el main estamos creando dos objetos, uno con variable de referencia p y otro con variable de referencia q; en las líneas 11 y 12 se le está asignando color y edad al objeto p pero en la línea 15, solo una variable de referencia p está apuntando a la variable de referencia q, no al objeto de q.

Por lo cual se inicializan los valores por defecto de los atributos. (No toma valores del método ya que este tampoco se ejecutó)

6. Which statements, when inserted independently into the following blank, will cause the code to print 2 at runtime? (Choose all that apply.)

```
int count = 0;
BUNNY: for(int row = 1; row <=3; row++)
    RABBIT: for(int col = 0; col <3 ; col++) {
        if((col + row) % 2 == 0)
            _____;
        count++;
    }
System.out.println(count);
```

A. break BUNNY

B. break RABBIT

C. continue BUNNY

D. continue RABBIT

E. break

F. continue

G. None of the above, as the code contains a compiler error

La respuesta correcta es B, C y E.

El ciclo for Bunny indica que la variable row inicializada en 1, debe ser ejecutada en el intervalo de cuando row sea menor o igual a 3 y esta en cada repetición se suma 1 a row.(ejecutándose por sí solo 3 veces)

Bunny ha "anidado" el ciclo for Rabbit indica la variable col inicializada en cero, ejecutando el ciclo cuando col sea menor a 3, avanzando de uno en uno. (ejecutándose 2 veces)

1+0 1

1+1 1

2+1 3

3+2 5

Así mismo este anido un condicional if que al sumar la variable col y row, al dividirse entre dos su residual es cero (es par) se ejecuta la línea de código a insertar y "else" se suma 1 a count.

A: "Break bunny" me haría salir del loop, y el valor de count quedaría en 1 por lo cual no es ese el valor.

B: "Break Rabbit" me haría solo ejecutar bunny, entonces al entrar al condicional la primera vuelta col+row sería 1, la segunda vuelta col+row 2, en el condicional se saldría al else y sumaría a count. terminando. Este si cumple "int count = 2";

C: "Continue Bunny" es igual que el anterior, solo ejecuta Bunny hasta que da el resultado.

D: "Continue Rabbit" reinicia rabbit con el valor de bunny, para después iniciar bunny con un nuevo valor, haciendo que count de un número más alto, en este caso "int count=5"

1+0 1

1+1 2 par

1+2 3

1+3 4 par

2+1 3

2+2 4 par

2+3 5

3+1 4 par

3+2 5

3+3 6 par

6. Which statements, when inserted independently into the following blank, will cause the code to print 2 at runtime? (Choose all that apply.)

```
int count = 0;
BUNNY: for(int row = 1; row <=3; row++)
    RABBIT: for(int col = 0; col <3 ; col++) {
        if((col + row) % 2 == 0)
            _____;
        count++;
    }
System.out.println(count);
```

A. break BUNNY

B. break RABBIT •

C. continue BUNNY

D. continue RABBIT

E. break

F. continue

G. None of the above, as the code contains a compiler error

E. **“Break”** es el equivalente a decir “breack rabbit” ya que al estar dentro de este, rompe al ciclo en el que se encuentra anidado el break.

“Continue” no afectaría, y daría el mismo resultado que continue Rabbit, siendo este “int count=5”

14. What is the output of the following code snippet?

```
2: double iguana = 0;
3: do {
4:     int snake = 1;
5:     System.out.print(snake++ + " ");
6:     iguana--;
7: } while (snake <= 5);
8: System.out.println(iguana);
```

- A. 1 2 3 4 -4.0
- B. 1 2 3 4 -5.0
- C. 1 2 3 4 5 -4.0
- D. 0 1 2 3 4 5 -5.0

E. The code does not compile.

F. The code compiles but produces an infinite loop at runtime.

G. None of the above

El código no compila por lo cual la respuesta correcta es la E, porque la variable local de "int snake = 1;" está definida dentro del ciclo, entonces si este se llega a ejecutar vuelve a inicializar en 1.