

Debug

Resolución de Katas en Java y Depuración

El propósito de esta actividad es fortalecer tus habilidades de programación en Java mediante la solución de tres problemas de katas y aprender a utilizar las herramientas de depuración en el entorno de desarrollo Eclipse.

Instrucciones

Solución de Katas

Resuelve los siguientes tres problemas (katas) en Java. Cada solución debe estar en un archivo Java independiente dentro del mismo proyecto de Eclipse:

Kata 1

[FizzBuzz](#)

Kata 2

[Tamagotchi v1.0](#)

Depuración en Eclipse

- Agrega **al menos dos puntos de interrupción (breakpoints)** en cada kata en las líneas del código que consideres clave para verificar la lógica.
- Utiliza el modo de depuración de Eclipse para ejecutar el programa y analizar el flujo de ejecución.
- Observa el comportamiento del programa en la consola y en las herramientas de depuración.

Entregables

1. Código Fuente:

Entrega los dos archivos Java con las soluciones a las katas, correctamente documentados y funcionales.

2. Capturas de Pantalla del Código:

Adjunta imágenes que muestren el código completo de cada kata en Eclipse.

3. Capturas de Pantalla del Proceso de Depuración:

- Al menos **dos capturas por kata** que muestren los puntos de interrupción en acción.
- Una captura debe mostrar el estado de las variables en el panel de depuración.

Criterios de Evaluación

- Soluciones correctas y funcionales para cada kata.
- Uso adecuado de puntos de interrupción y herramientas de depuración en Eclipse.
- Claridad y legibilidad del código entregado.
- Calidad de las capturas de pantalla (deben ser legibles y mostrar la información solicitada).

Fecha de Entrega: 15 de diciembre 2024

Formato de Entrega: Crea un repositorio (el repo debe ser público y tener un README explicando el contenido del proyecto) y sube los archivos de código como las capturas de pantalla y envía el enlace a la plataforma de la asignatura en Classroom.

Kata 1

FizzBuzz

```
public class FIZZBUZZ { // Declara una clase pública llamada FIZZBUZZ. Es la entrada principal de nuestro
programa.
    public FIZZBUZZ() { // Constructor de la clase FIZZBUZZ. Aunque no se usa en este programa, se define de
forma explícita.
    }

    public static void main(String[] var0) { // Método principal donde comienza la ejecución del programa.
        for(int var1 = 1; var1 <= 100; ++var1) { // Bucle `for` que recorre los números del 1 al 100.
            // La variable `var1` comienza en 1, se incrementa en 1 por cada iteración y termina cuando es mayor
que 100.

            if (var1 % 3 == 0 && var1 % 5 == 0) { // Comprueba si `var1` es divisible entre 3 Y entre 5 al mismo
tiempo.
                System.out.println("FIZBUZZ"); // Si la condición es verdadera, imprime "FIZBUZZ".
            } else if (var1 % 3 == 0) { // Si no es divisible entre ambos, verifica si es divisible solo entre
3.
                System.out.println("FIZZ"); // Si es divisible entre 3, imprime "FIZZ".
            } else if (var1 % 5 == 0) { // Si no es divisible entre 3, verifica si es divisible solo entre 5.
                System.out.println("BUZZ"); // Si es divisible entre 5, imprime "BUZZ".
            } else { // Si ninguna de las condiciones anteriores se cumple:
                System.out.println(var1); // Imprime el número actual (`var1`).
            }
        } // Cierra el bucle `for`. Se repite hasta que `var1` llega a 100.

    } // Cierra el método `main`.
} // Cierra la clase FIZZBUZZ.
```

You, 6 days ago | 1 author (You)

```
1 public class FIZZBUZZ {
2
3     Run | Debug
4     public static void main(String[] args) {
5         for (int i = 1; i<=100; i++) {
6             if(i%3==0 && i%5==0) {
7                 System.out.println(x:"FIZBUZZ");
8             }else if (i%3==0) {
9                 System.out.println(x:"FIZZ");
10
11
12             }else if (i%5==0) {
13                 System.out.println(x:"BUZZ");
14             }else {
15                 System.out.println(i);
16             }
17         }
18     }
19 }
```

You, 6 days ago • FIZZBUZZ

No Config

FIZZBUZZ.class

FIZZBUZZ.java

PrintStream.class

taamagoch

Run | Debug

FIZZBUZZ

VARIABLES

Local

args = String[0]@9

i = 6

WATCH

CALL STACK

PAUSED ON BREAK...

PAUSED ON BREAK...

FIZZBUZZ.main(Str

Thread [...] RUNNING

Thread [...] RUNNING

Thread [...] RUNNING

Thread [...] RUNNING

Thread [...] RUNNING

Thread [...] RUNNING

PAUSED ON BREAK...

PAUSED ON BREAK...

BREAKPOINTS

Uncaught Excep...

Caught Exceptio...

FIZZB... 8

FIZZBUZZ.java 12

AE-entorno > FIZZBUZZ.java > FIZZBUZZ > main(String[])

You, 6 days ago | 1 author (You)

1 public class FIZZBUZZ {

2

3 public static void main(String[] args) { args = String[0]@9

4 for (int i = 1; i<=100; i++) { i = 6

5 if(i%3==0 && i%5==0) { i = 6

6 System.out.println(x:"FIZZBUZZ");

7

8 }else if (i%3==0) { i = 6

9 System.out.println(x:"FIZZ");

10

11

12 }else if (i%5==0) { i = 6 You, 6 days ago * FIZZBUZZ

13 System.out.println(x:"BUZZ");

14 }else {

15 System.out.println(i);

16 }

17 }

18 }

19 }

PROBLEMS 1

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

GITLENS

DAW-1@DESKTOP-6JB57F8 MINGW64 ~/Desktop/Gerardo Corona López/debug fizzbug (master)

\$ cd c:\\Users\\DAW-1\\Desktop\\Gerardo\\Corona\\López\\debug\\fizzbug ; /usr/bin/env C:\\Program\\Files\\Java\\jdk-23\\bin\\java.exe -agentlib:jdwp=transport=dt_socket,server=n,suspend=y,address=localhost:63226 --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -cp C:\\Users\\DAW-1\\AppData\\Roaming\\Code\\User\\workspaceStorage\\df3fc88a298519f406bf62a813582877\\redhat.java\\jdt_ws\\debug\\fizzbug_9dbf2b62\\bin FIZZBUZZ

1

2

FIZZ

4

BUZZ

FIZZ

No Config

FIZZBUZZ.class

FIZZBUZZ.java

PrintStream.class

taamagoch

Run | Debug

FIZZBUZZ

VARIABLES

Local

args = String[0]@9

i = 30

WATCH

CALL STACK

PAUSED ON BREAK...

PAUSED ON BREAK...

FIZZBUZZ.main(Str

Thread [...]

Thread [...]

Thread [...]

Thread [...]

Thread [...]

Thread [...]

PAUSED ON BREAK...

PAUSED ON BREAK...

BREAKPOINTS

Uncaught Excep...

Caught Exceptio...

FIZZBUZZ.java

FIZZ...

AE-entorno > FIZZBUZZ.java > FIZZBUZZ > main(String[])

You, 6 days ago | 1 author (You)

1 public class FIZZBUZZ {

2

3 public static void main(String[] args) { args = String[0]@9

4 for (int i = 1; i<=100; i++) { i = 30

5 if(i%3==0 && i%5==0) { i = 30

6 System.out.println(x:"FIZZBUZZ");

7

8 }else if (i%3==0) { i = 30 You, 6 days ago • FIZZBUZZ

9 System.out.println(x:"FIZZ");

10

11

12 }else if (i%5==0) {

13 System.out.println(x:"BUZZ");

14 }else {

15 System.out.println(i);

16 }

17 }

18 }

19 }

PROBLEMS 1

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

GITLENS

BUZZ

FIZZ

22

23

FIZZ

26

FIZZ

28

29

FIZZBUZZ

bash

Debu...

Debu...

Kata 2

[Tamagotchi v1.0](#)

```
public class taamagochi { // Declara una clase pública llamada "taamagochi". Representa una mascota virtual.

    private int hunger; // Variable privada que mide el nivel de hambre (a mayor valor, más hambre tiene).

    private int energy; // Variable privada que mide el nivel de energía (a menor valor, más cansado está).

    private int mood;    // Variable privada que mide el nivel de ánimo o humor (a mayor valor, más feliz
está).

    public taamagochi() { // Constructor de la clase "taamagochi". Se ejecuta al crear una nueva instancia.

        this.hunger = 4; // Inicializa el nivel de hambre en 4.

        this.energy = 4; // Inicializa el nivel de energía en 4.

        this.mood = 4;    // Inicializa el nivel de humor en 4.

    }

    public String play() { // Método que simula que la mascota juega.
```

```
hunger++;          // Aumenta el nivel de hambre al jugar.

energy--;          // Disminuye el nivel de energía al jugar.

mood++;            // Aumenta el nivel de humor al jugar.

return getState(); // Devuelve el estado actual de la mascota después de jugar.
}

public String eat() { // Método que simula que la mascota come.

    hunger--;       // Disminuye el nivel de hambre.

    hunger--;       // Disminuye nuevamente el nivel de hambre (la comida satisface más).

    energy--;       // Disminuye el nivel de energía debido a la digestión.

    return getState(); // Devuelve el estado actual de la mascota después de comer.
}

public String sleep() { // Método que simula que la mascota duerme.

    energy++;       // Aumenta el nivel de energía al dormir.

    energy++;       // Aumenta nuevamente el nivel de energía (descanso profundo).
```



```
        return getState(); // Devuelve el estado actual de la mascota después de dormir.
    }

    public String getState() { // Método que determina el estado actual de la mascota basado en sus atributos.

        if (energy == 0) {           // Si la energía es 0, la mascota está durmiendo profundamente.

            return "(_ _) zZZ";      // Devuelve este estado para indicar sueño.

        } else if (mood > 8) {       // Si el nivel de humor es mayor a 8, la mascota está muy feliz.

            return ":-)";            // Devuelve este estado para mostrar felicidad.

        } else if (mood < 2) {       // Si el nivel de humor es menor a 2, la mascota está molesta o triste.

            return "ಠ_ಠ";            // Devuelve este estado para indicar disgusto o tristeza.

        } else if (energy < 2) {     // Si la energía es menor a 2, la mascota está cansada.

            return "(_ _)";          // Devuelve este estado para indicar cansancio.

        } else {                    // Si ninguna de las condiciones anteriores se cumple:

            return ":-|";            // Devuelve este estado para mostrar neutralidad.

        }

    }

}
```

```
}
```

```
public class tamaMain{  
    public static void main(String[]args){  
        taamagochi tamal = new taamagochi();  
  
        System.out.println(tamal.play());  
        System.out.println(tamal.eat());  
        System.out.println(tamal.sleep());  
    }  
}
```

No Config

FIZZBUZZ.class

taamagochi.java 1, U

AE-entorno > taamagochi.java > taamagochi > play()

Local

this = taamagochi...

1

public class taamagochi {

7

8

9

10

11 public taamagochi() {

12 this.hunger=4;

13 this.energy=4;

14 this.mood=4;

15 }

16

17 public String play(){

18 hunger++; hunger = 5

19 energy--; energy = 3

20 mood++; mood = 5

21

22 return getState();

23

24

25 }

26

27 public String eat() {

28 hunger--;

29 hunger--;

30 energy--;

31 return getState();

32

33 }

34

35 public String sleep() {

36 energy++;

37 energy++;

WATCH

CALL STACK

BREAKPOINTS

Uncaught Excep...

Caught Excep...

taam...

taamagochi...

tamaMain.java

tamaMain.java

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

GITLENS

DAW-1@DESKTOP-6JB57F8 MINGW64 ~/Desktop/Gerardo Corona López/debug fizzbug (master)

\$ /usr/bin/env C:\Program Files\Java\jdk-23\bin\java.exe -agentlib:jdwp=transport=dt_socket,server=n,suspend=y,address=localhost:63419 --enable-preview -XX:+ShowCodeDetailsInExceptionMessag

es -cp C:\Users\DAW-1\AppData\Roaming\Code\User\workspaceStorage\df3fc88a298519f406bf62a813582877\redhat.java\jdt_ws\debug\ fizzbug_9dbf2b62\bin tamaMain

DAW-1@DESKTOP-6JB57F8 MINGW64 ~/Desktop/Gerardo Corona López/debug fizzbug (master)

\$ cd c:\Users\DAW-1\Desktop\Gerardo Corona López\debug\ fizzbug ; /usr/bin/env C:\Program Files\Java\jdk-23\bin\java.exe -agentlib:jdwp=transport=dt_socket,server=n,suspend=y,address

=localhost:63480 --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -cp C:\Users\DAW-1\AppData\Roaming\Code\User\workspaceStorage\df3fc88a298519f406bf62a813582877\redhat.java\jdt_w

s\debug\ fizzbug_9dbf2b62\bin tamaMain

No Config

FIZZBUZZ.class

FIZZBUZZ.java M

taamagochi.java 1, U

tamaMain.java M

Debug.pdf U

VARIABLES

Local

this = taamagochi...

AE-entorno > taamagochi.java > taamagochi > play()

1 public class taamagochi { // Declara una clase pública llamada "taamagochi". Representa una mascota virtual.

2

3 private int hunger; // Variable privada que mide el nivel de hambre (a mayor valor, más hambre tiene).

4 private int energy; // Variable privada que mide el nivel de energía (a menor valor, más cansado está).

5 private int mood; // Variable privada que mide el nivel de ánimo o humor (a mayor valor, más feliz está).

6

7 public taamagochi() { // Constructor de la clase "taamagochi". Se ejecuta al crear una nueva instancia.

8 this.hunger = 4; // Inicializa el nivel de hambre en 4.

9 this.energy = 4; // Inicializa el nivel de energía en 4.

10 this.mood = 4; // Inicializa el nivel de humor en 4.

11 }

12

13 public String play() { // Método que simula que la mascota juega.

14 hunger++; // Aumenta el nivel de hambre al jugar. hunger = 5

15 energy--; // Disminuye el nivel de energía al jugar. energy = 4

16 mood++; // Aumenta el nivel de humor al jugar.

17 return getState(); // Devuelve el estado actual de la mascota después de jugar.

18 }

19

20 public String eat() { // Método que simula que la mascota come.

21 hunger--; // Disminuye el nivel de hambre.

22 hunger--; // Disminuye nuevamente el nivel de hambre (la comida satisface más).

23 energy--; // Disminuye el nivel de energía debido a la digestión.

24 return getState(); // Devuelve el estado actual de la mascota después de comer.

25 }

26

27 public String sleep() { // Método que simula que la mascota duerme.

28 energy++; // Aumenta el nivel de energía al dormir.

29 energy++; // Aumenta nuevamente el nivel de energía (descanso profundo).

30 return getState(); // Devuelve el estado actual de la mascota después de dormir.

31 }

32

WATCH

CALL STACK

BREAKPOINTS

Uncaught Excep...

Caught Exceptio...

taamagochij... 15

taam... 46

PROBLEMS 1

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

GITLENS

\$ cd c:\\Users\\DAW-1\\Desktop\\Gerardo\\ Corona\\ López\\debug\\ fizzbug ; /usr/bin/env C:\\Program\\ Files\\Java\\jdk-23\\bin\\java.exe -agentlib:jwp=transport=dt_socket,server=n,suspend=y,address=localhost:63807 --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -cp C:\\Users\\DAW-1\\AppData\\Roaming\\Code\\User\\workspaceStorage\\df3fc88a298519f406bf62a813582877\\redhat.java\\jdt_w s\\debug\\ fizzbug_9dbf2b62\\bin tamaMain

:-|

:-|

:-|

DAW-1@DESKTOP-6JB57F8 MINGW64 ~/Desktop/Gerardo Corona López/debug fizzbug (master)

\$ cd c:\\Users\\DAW-1\\Desktop\\Gerardo\\ Corona\\ López\\debug\\ fizzbug ; /usr/bin/env C:\\Program\\ Files\\Java\\jdk-23\\bin\\java.exe -agentlib:jwp=transport=dt_socket,server=n,suspend=y,address=localhost:63829 --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -cp C:\\Users\\DAW-1\\AppData\\Roaming\\Code\\User\\workspaceStorage\\df3fc88a298519f406bf62a813582877\\redhat.java\\jdt_w s\\debug\\ fizzbug_9dbf2b62\\bin tamaMain