Fecha: 22/03/2019
Akurey-General Training
Camila Viquez Alpizar
Asignación 01 - Fragmentos de código

Extraiga 5 ejemplos de código donde haya aplicado buenas prácticas de programación

- Meaningful Names

Bloque de codigo

```
public class Facturar implements ActionListener  {
   private Factura factura;
   private Producto producto;
   private Usuario usuario;
   private inventario vistaInventario;
   private listarVentas vistaListarVentas;
   private opciones vistaOpciones;
   private inicio vistaInicio;


   public Facturar() {
   }
   public Facturar(Factura factura, Producto producto, Usuario usuario, inicio
vistaInicio, inventario vistaInventario, listarVentas vistaListarVentas, opciones
vistaOpciones) {
       this.factura = factura;
       this.usuario = usuario;
       this.producto = producto;
       this.vistaInicio = vistaInicio;
       this.vistaInventario = vistaInventario;
       this.vistaOpciones = vistaOpciones;
       this.vistaListarVentas = vistaListarVentas;
       this.btnActions();


   }
```

Bloque de codigo [1]

- Functions

Bloque de codigo

```
def poliganTriangulation(numTriangles, numDiagonal, firstVertex):
	diagonalList = []
	for i in range(numDiagonal):
		drawDiagonalIrregular(firstVertex, diagonalList,i+3)
		drawDiagonal(firstVertex, diagonalList,i+3)

def drawDiagonal(firstVertex, diagonalList,i):
	diagonalList.append([firstVertex, i])
	return diagonalList

def drawDiagonalIrregular(firstVertex, diagonalList,i):
	 distance = random.randint(1,20)
	 print("Irregular polygon Distance between one vertex to another: " +
str(distance))

def factorial(number):
	if(number > 0):
		n = number-1
		while(n>0):
			number *= n
			n -= 1
	return(number)
```

Bloque de codigo [2]

- Formatting

Bloque de codigo

```
def poliganTriangulation(numTriangles, numDiagonal, firstVertex):
	diagonalList = []
	for i in range(numDiagonal):
		drawDiagonalIrregular(firstVertex, diagonalList,i+3)
		drawDiagonal(firstVertex, diagonalList,i+3)

def drawDiagonal(firstVertex, diagonalList,i):
	diagonalList.append([firstVertex, i])
	return diagonalList

def drawDiagonalIrregular(firstVertex, diagonalList,i):
	 distance = random.randint(1,20)
	 print("Irregular polygon Distance between one vertex to another: " +
str(distance))

def factorial(number):
	if(number > 0):
		n = number-1
		while(n>0):
			number *= n
			n -= 1
	return(number)
```

Bloque de codigo [3]

- Classes

Bloque de codigo

```
public class Controlador {
     private DTODinamica dtoDinamica;

    public Controlador(DTODinamica capsulaDinamica) {
        this.dtoDinamica = capsulaDinamica;
        System.out.println(dtoDinamica.toString());
    }


     public DTODinamica procesarConsultaDinamica(DTODinamica capsulaDinamica){


         return capsulaDinamica;
     }
}
```

Bloque de codigo [4]

- Meaningful Names

Bloque de codigo

```
public double verificarDescuento(int total){
    System.out.println("totalIngresando:"+String.valueOf(total));
    if(validarEstudiante()){

        double nuevoMonto = total *0.1;
        return nuevoMonto;
    }else{
        return 0;
    }

}
```

Bloque de codigo [5]

Fecha: 22/03/2019
Akurey-General Training
Camila Viquez Alpizar
Asignación 01 - Fragmentos de código

Extraiga 5 ejemplos de código donde no haya aplicado buenas prácticas de programación
A continuación se muestra una secuencia bloques de código que tienen la finalidad de representar la falta de buenas practicas de programacion. Se encuentra subdividido en cuatro aspectos: meaningful names, functions, format y classes.

- Meaningful Names

Bloque de codigo

```
public abstract class Camisa implements Cloneable {
    private String color;
    private String talla;
    private String estampado;

    protected Camisa(String color, String talla, String estampado){
        setColor(color);
        setTalla(talla);
        setEstampado(estampado);
    }
```

Mejora del bloque

```
public abstract class Camisa implements Cloneable {
    private String colorCamisa;
    private String tallaCamisa;
    private String estampadoCamisa;

    protected Camisa(String color, String talla, String estampado){
        setColorCamisa(color);
        setTallaCamisa(talla);
        setEstampadoCamisa(estampado);
    }
```

Bloque de codigo [6]

Bloque de codigo

```
def funcPrincipal(tablero,listCasillas):
    aux_tablero = []
    for i in range(0,len(tablero)):
        contB = 0 #cuantos son por fila blancos
```

```
        indice = 0 #guardas la posición negro
        valorFila = 0
        listaPoda = []
        for j in range(0,len(tablero[0])):
            if(tablero[i][j][0] == "negra"):
                if(j < len(tablero[0])-1):
                    valorFila = tablero[i][j][1]
            else:
                valorColumna = getSuma(tablero,i,j)
                listaPoda =poda(tablero[i][j],valorFila,valorColumna)
                bkt = backTracking(combinaciones,aux_tablero,tablero)
    printMatriz(bkt)
    return bkt
```

Mejora del bloque

```
contB = 0 #cuantos son por fila blancos
La variable podría tener un nombre más significativo cmo contadorBlancos o
contCuadrosBlancos y asi no necesitaría de un comentario aclaratorio
```

Bloque de codigo [7]

- Functions

Bloque de codigo

```
public abstract class Camisa implements Cloneable {
    private String color;
    private String talla;
    private String estampado;

    protected Camisa(String color, String talla, String estampado){
        setColor(color);
        setTalla(talla);
        setEstampado(estampado);
    }
```

Mejora del bloque

```
public abstract class Camisa implements Cloneable {
    private String colorCamisa;
    private String tallaCamisa;
    private String estampadoCamisa;
```

```
    protected Camisa(String color, String talla, String estampado){
        setColorCamisa(color);
        setTallaCamisa(talla);
        setEstampadoCamisa(estampado);
    }
```

Bloque de codigo [8]

- Formatting

Bloque de codigo

```
def crearListaPosibles(rango):
    if(rango > 0):
        lista = []
        num = 1
        while(num <= rango-1 and num <= 9):
            lista.append(num)
            num += 1
        return lista

def poda(posicion, debeSumarF, debeSumarC):
        l_posiblesF = crearListaPosibles(debeSumarF)
        conj_posiblesF = set(l_posiblesF)

        l_posiblesC = crearListaPosibles(debeSumarC)
        conj_posiblesC = set(l_posiblesC)

        interConj = conj_posiblesF.intersection(conj_posiblesC)
        listInterConj = list(interConj)
        return listInterConj
```

Mejora del bloque

```
def poda(posicion, debeSumarF, debeSumarC):
        l_posiblesF = crearListaPosibles(debeSumarF)
        conj_posiblesF = set(l_posiblesF)

        l_posiblesC = crearListaPosibles(debeSumarC)
        conj_posiblesC = set(l_posiblesC)

        interConj = conj_posiblesF.intersection(conj_posiblesC)
        listInterConj = list(interConj)
        return listInterConj
```

```
def crearListaPosibles(rango):
    if(rango > 0):
        lista = []
        num = 1
        while(num <= rango-1 and num <= 9):
            lista.append(num)
            num += 1
        return lista

La función crearLista() es llamada por la función poda() por lo que debería estar
abajo de esta.
```

Bloque de codigo [9]

- Classes

Bloque de codigo

```
public Facturar() {
   }
   public Facturar(Factura factura, Producto producto, Usuario usuario, inicio
vistaInicio, inventario vistaInventario, listarVentas vistaListarVentas, opciones
vistaOpciones) {
       this.factura = factura;
       this.usuario = usuario;
       this.producto = producto;
       this.vistaInicio = vistaInicio;
       this.vistaInventario = vistaInventario;
       this.vistaOpciones = vistaOpciones;
       this.vistaListarVentas = vistaListarVentas;
       this.btnActions();


   }

   //listas de invetario y usuarios

   private ArrayList<Producto> inventarioLista = new  ArrayList<Producto>();
   private ArrayList<Usuario> estudiantes = new  ArrayList<Usuario>();
   private ArrayList<Usuario> funcionarios = new  ArrayList<Usuario>();
   private ArrayList<Usuario> usuarios = new  ArrayList<Usuario>();
   ArrayList<Producto> productosRegistrados = new  ArrayList<Producto>();
   ArrayList<Producto> ventasRegistradas = new  ArrayList<Producto>();
```

```
int totalV =0;
```

Mejora del bloque

```
 private ArrayList<Producto> inventarioLista = new  ArrayList<Producto>();
   private ArrayList<Usuario> estudiantes = new  ArrayList<Usuario>();
   private ArrayList<Usuario> funcionarios = new  ArrayList<Usuario>();
   private ArrayList<Usuario> usuarios = new  ArrayList<Usuario>();
   ArrayList<Producto> productosRegistrados = new  ArrayList<Producto>();
   ArrayList<Producto> ventasRegistradas = new  ArrayList<Producto>();
int totalV =0;

public Facturar() {
   }
   public Facturar(Factura factura, Producto producto, Usuario usuario, inicio
vistaInicio, inventario vistaInventario, listarVentas vistaListarVentas, opciones
vistaOpciones) {
       this.factura = factura;
       this.usuario = usuario;
       this.producto = producto;
       this.vistaInicio = vistaInicio;
       this.vistaInventario = vistaInventario;
       this.vistaOpciones = vistaOpciones;
       this.vistaListarVentas = vistaListarVentas;
       this.btnActions();
}

Los atributos deben estar declarados antes de el constructor de la clase
```

Bloque de codigo [10]

Fecha: 22/03/2019
Akurey-General Training
Camila Viquez Alpizar
Asignación 01 - Fragmentos de código

Referencias


[1] Tec digital:
https://tecdigital.tec.ac.cr/dotlrn/classes/CA/IC5821/S-1-2018.CA.IC5821.2/evaluation/tda-ce-estudiante/tda-index?page_num=3

[2] Repositorio git:
https://github.com/camilaViquez/PoliganTriangulation/blob/develop/poligonTriangulation.py

[3] Repositorio git:
https://github.com/camilaViquez/PoliganTriangulation/blob/develop/poligonTriangulation.py

[4] Repositorio git:
https://github.com/camilaViquez/PatronesComportamiento-Incidentes/blob/master/src/controlador/Controlador.java

[5] Tec digital:
https://tecdigital.tec.ac.cr/dotlrn/classes/CA/IC6821/S-2-2018.CA.IC6821.2/evaluation/tda-ce-estudiante/tda-index?page_num=3

[6] Repositorio git:
https://github.com/lmata98/ExamenII/blob/master/src/Caso2Prototype/Camisa.java

[7] Repositorio git: https://github.com/camilaViquez/kakuro/blob/master/kakuros.py

[8] Repositorio git: https://github.com/camilaViquez/kakuro/blob/master/kakuros.py

[9] Repositorio git: https://github.com/camilaViquez/kakuro/blob/master/kakuros.py

[10] Tec digital:
https://tecdigital.tec.ac.cr/dotlrn/classes/CA/IC6821/S-2-2018.CA.IC6821.2/evaluation/tda-ce-estudiante/tda-index?page_num=3