```
# encoding: utf-8

import gvsig
from gvsig import currentLayer
from gvsig import currentView
from gvsig import createFeatureType
from gvsig import createShape
from gvsig import getResource

from gvsig.geom import POLYGON
from gvsig.geom import POINT, MULTIPOINT
from gvsig.geom import D2
from gvsig.geom import createPoint

from gvsig.commonsdialog import msgbox

from gvsig.libs.formpanel import FormPanel

"""
Notas asocidas al taller de introduccion a scripting.

Para estos ejercicios precisaremos:

- Crear una vista en 23030
- Cargarenos la capa manzanas_pob de
  gvSIG/extensiones/org.gvsig.scripting.app.mainplugin/scripting/scripts/data
- Cargaremos la capa manzanas_puntos
- Seleccionaremos la capa de puntos

"""

def listadoDePuntos():
  """
  Como recorrer los elementos de una capa, acceder
  a sus atributos y su geometria

  Precisa que este activa la capa de puntos.
  """
  # Obtenemos la capa de puntos que debera estar seleccionada
  layer = currentLayer()
  print layer.getName()
  # Nos recorremos las features de la capa de puntos
  features = layer.features()
  for f in features:
    # Para cada feature recuperamos su id, y la X e Y de su geometria
    print f.get("ID"), f.getDefaultGeometry().getX(), f.getDefaultGeometry().getY()

def crearShapeConBuffer():
  """
  Como crear una capa de poligonos a partir de una capa de puntos en
  base a crear un area de influencia para cada punto

  Precisa que este activa la capa de puntos.
  """
  schema = createFeatureType()

  schema.append("GEOMETRY", "GEOMETRY")
  schema.get("GEOMETRY").setGeometryType(POLYGON, D2)
  schema.append("ID","INTEGER",10)
  output = createShape(schema,prefixname="buffer")

  layer = currentLayer()
  print layer.getName()
  #Iniciamos edicion en la capa de salida
  outputstore = output.getFeatureStore()
  outputstore.edit()
  for feature in layer.features():
    geom = feature.getDefaultGeometry()
    # Para cada featura de entrada creamos una de salida
    # y copiamos los datos de origen en ella
```

```
    newfeature = outputstore.createNewFeature()
    newfeature.set("ID",feature.get("ID"))
    # A la geoemtria le aplicamos un buffer
    newfeature.setDefaultGeometry(geom.buffer(50))
    # y insertamos la nueva feature en la capa de salida
    outputstore.insert(newfeature)

  # Al terminar de procesar las features, terminamos la edicion
  # en la capa de salida para consolidar los datos.
  outputstore.finishEditing()

  currentView().addLayer(output)


def crearShapeConBufferYDistancia():
  """
  Como crear una capa de poligonos a partir de una capa de puntos en
  base a crear un area de influencia para cada punto.
  Ademas calcularemos para cada punto la distancia de este a un
  punto dado y la guardaremos en una columna de la nueva capa de
  poligonos.

  Precisa que este activa la capa de puntos.
  """
  schema = createFeatureType()

  schema.append("GEOMETRY", "GEOMETRY")
  schema.get("GEOMETRY").setGeometryType(POLYGON, D2)

  #+1
  schema.append("DISTANCIA","DOUBLE",10)

  schema.append("ID","INTEGER",10)
  output = createShape(schema,prefixname="buffer")

  #+3
  refpoint = createPoint(D2)
  refpoint.setX(725000)
  refpoint.setY(4370000)

  layer = currentLayer()
  print layer.getName()
  #Iniciamos edicion en la capa de salida
  outputstore = output.getFeatureStore()
  outputstore.edit()
  for feature in layer.features():
    geom = feature.getDefaultGeometry()
    # Para cada featura de entrada creamos una de salida
    # y copiamos los datos de origen en ella
    newfeature = outputstore.createNewFeature()
    newfeature.set("ID",feature.get("ID"))

    #+1
    newfeature.set("DISTANCIA",geom.distance(refpoint))

    # A la geoemtria le aplicamos un buffer
    newfeature.setDefaultGeometry(geom.buffer(50))
    # y insertamos la nueva feature en la capa de salida
    outputstore.insert(newfeature)

  # Al terminar de procesar las features, terminamos la edicion
  # en la capa de salida para consolidar los datos.
  outputstore.finishEditing()

  currentView().addLayer(output)
  # Aplicar una leyenda de intervalos para ver el resultado
  # de forma grafica

def crearShapeConBufferYDistanciaFiltrando():
  """
```

```
  Como crear una capa de poligonos a partir de una capa de puntos en
  base a crear un area de influencia para cada punto.
  Calcularemos para cada punto la distancia de este a un
  punto dado y la guardaremos en una columna de la nueva capa de
  poligonos.
  Solo operaremos con los puntos que intersecten con la capa de
  manzanas ignorando los puntos que no intersecten.

  Precisa que este activa la capa de puntos.
  """
  schema = createFeatureType()

  schema.append("GEOMETRY", "GEOMETRY")
  schema.get("GEOMETRY").setGeometryType(POLYGON, D2)
  schema.append("DISTANCIA","DOUBLE",10)
  schema.append("ID","INTEGER",10)
  output = createShape(schema,prefixname="buffer")

  refpoint = createPoint(D2)
  refpoint.setX(725000)
  refpoint.setY(4370000)

  #+2
  # Obtenemos una referencia a la capa de manzanas
  manzanas = currentView().getLayer("manzanas_pob")

  layer = currentLayer()
  print layer.getName()
  features = layer.features()
  #Iniciamos edicion en la capa de salida
  outputstore = output.getFeatureStore()
  outputstore.edit()
  for feature in features.iterator():
    geom = feature.getDefaultGeometry()
    #+3 y sangrar
    for manzana in manzanas.features():
      geom_manzana = manzana.getDefaultGeometry()
      # En el siguiente "if" podriamos añadir mas condiciones
      # de filtro, por ejemplo solo para las manzanas con una
      # determinada poblacion
      if geom.intersects(geom_manzana): # and manzana.get("pob_total") > 100:
        # Para cada featura de entrada creamos una de salida
        newfeature = outputstore.createNewFeature()
        newfeature.set("ID",feature.get("ID"))
        newfeature.set("DISTANCIA",geom.distance(refpoint))
        # A la geoemtria le aplicamos un buffer
        newfeature.setDefaultGeometry(geom.buffer(50))
        # y insertamos la nueva feature en la capa de salida
        outputstore.insert(newfeature)

  # Al terminar de procesar las features, terminamos la edicion
  # en la capa de salida para consolidar los datos.
  outputstore.finishEditing()

  currentView().addLayer(output)
  # Aplicar una leyenda de intervalos para ver el resultado
  # de forma grafica

def crearShapeDesplazado():
  """
  Vamos a crear una nueva capa de puntos copiando los puntos de la capa
  de puntos (manzanas_puntos) y desplazando estos 100 en cada eje (x e y).

  Precisa que este activa la capa de puntos.
  """
  schema = createFeatureType()

  schema.append("GEOMETRY", "GEOMETRY")
  schema.get("GEOMETRY").setGeometryType(POINT, D2)
  schema.append("ID","INTEGER",10)
```

```
  output = createShape(schema, prefixname="shift")

  layer = currentLayer()
  print layer.getName()

  outputstore = output.getFeatureStore()
  outputstore.edit()
  for feature in layer.features():
    geom = feature.getDefaultGeometry()

    newfeature = outputstore.createNewFeature()
    newfeature.set("ID",feature.get("ID"))
    point = createPoint(D2)
    point.setX(geom.getX()+100)
    point.setY(geom.getY()+100)
    newfeature.setDefaultGeometry(point)

    outputstore.insert(newfeature)

  outputstore.finishEditing()

  currentView().addLayer(output)

def crearShapeDesplazado2(layer, despX, despY):
  """
  Vamos a crear una funcion, que reciba una capa, y un desplazamiento
  para X e Y, y crea una nueva capa de puntos copiando los puntos de la capa
  de puntos que recibe como parametro y desplazando estos la cantidad especificada
  en el parametro despX e despY.
  """
  if layer == None:
    msgbox("Debera indicar la capa origen")
    return

  schema = createFeatureType()

  schema.append("GEOMETRY", "GEOMETRY")
  schema.get("GEOMETRY").setGeometryType(POINT, D2)
  schema.append("ID","INTEGER",10)
  output = createShape(schema, prefixname="shift")

  outputstore = output.getFeatureStore()
  outputstore.edit()
  id=1
  for feature in layer.features():
    geom = feature.getDefaultGeometry()

    newfeature = outputstore.createNewFeature()
    newfeature.set("ID",id)
    point = createPoint(D2)
    point.setX(geom.centroid().getX()+despX)
    point.setY(geom.centroid().getY()+despY)
    newfeature.setDefaultGeometry(point)
    id+=1
    outputstore.insert(newfeature)

  outputstore.finishEditing()
  currentView().addLayer(output)

#
# -------------------------
#

from org.gvsig.tools.swing.api import ToolsSwingLocator
from org.gvsig.app import ApplicationLocator

def setLayersComboBoxModel(combo):
  """
  Esta funcion recibe un JComboBox y lo rellena con la lista de
  capas que hay cargadas en las vistas del proyecto.
```

```
  """
  layersModel = ApplicationLocator.getManager().createProjectLayersTreeModel()
  ToolsSwingLocator.getToolsSwingManager().setTreeModel(combo,layersModel)

class DesplazarPanel(FormPanel):
  """
  Clase que presenta un formulario pidiendo que se seleccione una capa
  y un desplazamiento para X e Y y genera una nueva capa desplazando los
  puntos de la capa original la cantidad especificada.

  Usar la opcion de menus:

    Herramientas -> Abeille forms designer

  Para acceder al editor de formularios y abrir "desplazarPanel.xml"
  para ver y editar el formulario asociado a este codigo.
  """
  def __init__(self):
    FormPanel.__init__(self,getResource(__file__,"desplazarPanel.xml"))
    setLayersComboBoxModel(self.cboLayers)
    self.setPreferredSize(300,150)

  def btnCancelar_click(self,*args):
    self.hide()

  def btnDesplazar_click(self,*args):
    try:
      despX = int(self.txtDesplazamientoX.getText())
      despY = int(self.txtDesplazamientoY.getText())
    except:
      msgbox("Valores incorrectos para desplazamiento X o y")
      return
    layer = self.cboLayers.getSelectedItem()
    if layer == None:
      msgbox("Debera seleccionar una capa")
      return
    msgbox("Desplazar la capa %r, desplazamiento X %s, desplazamiento y %s" % (
        layer.getName(),
        despX,
        despY
      )
    )
    crearShapeDesplazado2(layer, despX, despY)


#
# ------------------------
#

def main(*args):
  """
  Punto de entrada del script.

  Aqui se llaman a las distintas funciones que ilustran los
  ejercicios realizados en el taller.

  Descomentarizar la funcion que interese (eliminando el # al inicio
  de la linea) para probar el codigo.

  """
  #listadoDePuns()
  #crearShapeConBuffer()
  #crearShapeConBufferYDistancia()
  #crearShapeConBufferYDistanciaFiltrando()
  #crearShapeDesplazado()
  #crearShapeDesplazado2(currentView().getLayer("manzanas_puntos"), 200, 200)

  #panel = DesplazarPanel()
  #panel.showWindow("Desplazar")
```

```
print "Taller de introduccion a scripting"
```