**Завдання**

**Завдання**:

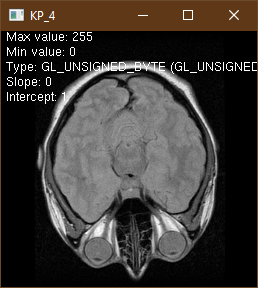
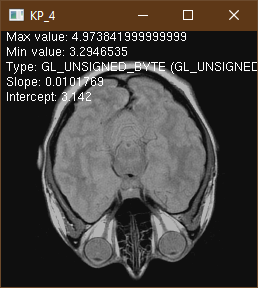
1. Вивчити теоретичні відомості щодо специфіки роботи з піксельними даними медичних зображень.
2. Розробити пограмний застосунок для завантаження медичного зображення в форматі DICOM та виконання операцій з піксельними даними цього зображення
3. Розміри частини вікна програмного застосунку для візуалізації графічних даних мають відповідати розмірам завантаженого медичного зображення, завантажене медичне зображення має мати масштаб 100%
4. Створити події при обробці яких можна виконати задану в варіанті завдання операцію, що призведе до зміни представленого в пам’яті погроми піксельних даних зображення та його відображення на екрані, та відновити оригінальне відображення завантаженого томографічного знімку
5. Визначити т вивести на екран інформацію стосовно піксельних даних: мінімальне та максимальне значення, що зберігаються в піксельних даних відображуваного зображення, значення коефіцієнта масштабування, значення коефіцієнта зсуву, типу пікселних даних для відображуваного зображення.
6. Виконати збереження створеного відповідно до заданого в варіанті завдання похідного зображення до файлу в форматі DICOM. Надати можливість завантаження збереженого похідного зображення розроблену пограму із відповідним оновленням інформації стосовно пікселних даних, що має виводитись на екан.
7. Скласти і захистити звіт.

| Номер варіанту | Коефіцієнт маштабування | Коефіцієнт зсуву | Верхня межа | Нижня межа |
| --- | --- | --- | --- | --- |
| 2 | 0.0101769 | 3,142 | 180 | 15 |

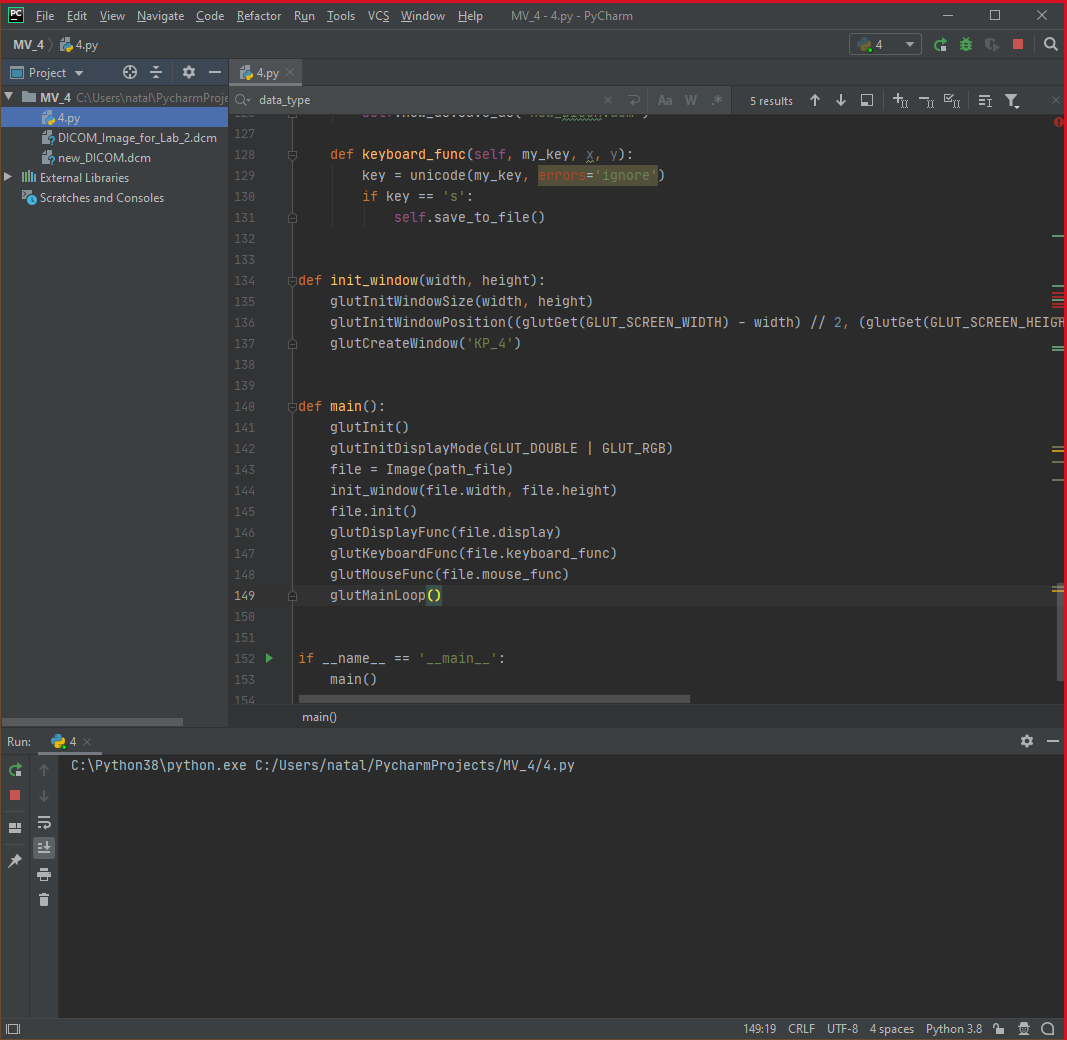
**Лістинг программи**

from OpenGL.GL import \*  
from OpenGL.GLU import \*  
from OpenGL.GLUT import \*  
import pydicom  
import numpy as np  
from pydicom.uid import generate\_uid  
  
path\_file = "DICOM\_Image\_for\_Lab\_2.dcm"  
  
  
class Image:  
 def \_\_init\_\_(self, path):  
 self.high\_border = 180  
 self.low\_border = 15  
 self.scaling\_factor = 0.0101769  
 self.shear\_ratio = 3.142  
 self.min\_el = 0  
 self.max\_el = 255  
  
 self.ds = pydicom.read\_file(path)  
 self.new\_ds = pydicom.read\_file(path)  
 self.intercept = self.ds[0x0281053].value  
 self.slope = self.ds[0x0281052].value  
  
 self.image\_pixels = self.ds.pixel\_array  
 self.new\_image\_pixels = np.array(self.image\_pixels).astype(float)  
  
 self.bits = self.ds[0x280100].value  
 self.data\_type = GL\_FLOAT if (self.slope != 0) & (self.intercept != 1) else GL\_UNSIGNED\_BYTE if self.bits == 8 else GL\_UNSIGNED\_INT  
 self.width, self.height = self.ds[0x280010].value, self.ds[0x280011].value  
 self.isTextVisible = True  
  
 def init(self):  
 glClearColor(0, 0, 0, 0.0)  
 glMatrixMode(GL\_PROJECTION)  
 glLoadIdentity()  
 gluOrtho2D(0, self.width, 0, self.height)  
  
 def display(self):  
 glClear(GL\_COLOR\_BUFFER\_BIT)  
 glColor3f(1, 1, 1)  
  
 self.draw\_texture(self.new\_image\_pixels, self.data\_type)  
 if self.isTextVisible:  
 self.print\_text(5, self.height - 10, GLUT\_BITMAP\_HELVETICA\_12, "Max value: " + str(self.max\_el))  
 self.print\_text(5, self.height - 25, GLUT\_BITMAP\_HELVETICA\_12, "Min value: " + str(self.min\_el))  
 self.print\_text(5, self.height - 40, GLUT\_BITMAP\_HELVETICA\_12, "Type: " + str(self.data\_type))  
 self.print\_text(5, self.height - 55, GLUT\_BITMAP\_HELVETICA\_12, "Slope: " + str(self.slope))  
 self.print\_text(5, self.height - 70, GLUT\_BITMAP\_HELVETICA\_12, "Intercept: " + str(self.intercept))  
 glutSwapBuffers()  
  
 def print\_text(self, x, y, font, line):  
 glColor3f(1, 1, 1)  
 glPushAttrib(GL\_DEPTH\_TEST)  
 glRasterPos2d(x, y)  
 for i in line:  
 glutBitmapCharacter(font, ord(i))  
 glPopAttrib()  
  
 def draw\_texture(self, data, data\_type):  
 glTexImage2D(GL\_TEXTURE\_2D, 0, GL\_LUMINANCE, self.width, self.height, 0, GL\_LUMINANCE, data\_type, data)  
 glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_MAG\_FILTER, GL\_NEAREST)  
 glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_MIN\_FILTER, GL\_NEAREST)  
 glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_WRAP\_S, GL\_CLAMP)  
 glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_WRAP\_T, GL\_CLAMP)  
 glEnable(GL\_TEXTURE\_2D)  
  
 glBegin(GL\_QUADS)  
 glTexCoord2d(0.0, 0.0)  
 glVertex2d(0.0, 0.0)  
 glTexCoord2d(1.0, 0.0)  
 glVertex2d(self.width, 0.0)  
 glTexCoord2d(1.0, 1.0)  
 glVertex2d(self.width, self.height)  
 glTexCoord2d(0.0, 1.0)  
 glVertex2d(0.0, self.height)  
 glEnd()  
 glDisable(GL\_TEXTURE\_2D)  
  
 def change\_value(self):  
 for i in range(self.height):  
 for j in range(self.width):  
 if self.new\_image\_pixels[i, j] > self.high\_border:  
 self.new\_image\_pixels[i, j] = float(self.high\_border \* self.scaling\_factor + self.shear\_ratio)  
 elif self.new\_image\_pixels[i, j] < self.low\_border:  
 self.new\_image\_pixels[i, j] = float(self.low\_border \* self.scaling\_factor + self.shear\_ratio)  
 else:  
 self.new\_image\_pixels[i, j] = float(self.new\_image\_pixels[i, j] \* self.scaling\_factor + self.shear\_ratio)  
 self.min\_el, self.max\_el = self.new\_image\_pixels.min(), self.new\_image\_pixels.max()  
 for i in range(self.height):  
 for j in range(self.width):  
 self.new\_image\_pixels[i, j] = ((self.new\_image\_pixels[i, j] - self.min\_el) \* (self.high\_border - self.low\_border)) / (self.max\_el - self.min\_el) + self.low\_border  
  
 def return\_value(self):  
 self.new\_image\_pixels = np.array(self.image\_pixels).astype(float)  
  
 def mouse\_func(self, button, state, x, y):  
 if button == GLUT\_LEFT\_BUTTON and state == GLUT\_DOWN:  
 self.slope = self.scaling\_factor  
 self.intercept = self.shear\_ratio  
 self.change\_value()  
 self.display()  
 if button == GLUT\_RIGHT\_BUTTON and state == GLUT\_DOWN:  
 self.max\_el = 255  
 self.min\_el = 0  
 self.intercept = self.ds[0x0281053].value  
 self.slope = self.ds[0x0281052].value  
 self.return\_value()  
 self.display()  
  
 def save\_to\_file(self):  
 self.new\_ds.PixelData = self.new\_image\_pixels.tobytes()  
 self.new\_ds[0x0281052].value = self.slope  
 self.new\_ds[0x0281053].value = self.intercept  
 self.new\_ds[0x0080008].value = ('DERIVED', 'SECONDARY')  
 self.new\_ds[0x008103e].value = 'slope-{0}, intercept-{1}'.format(self.slope, self.intercept)  
 if 'SeriesDescriptionCode' in self.new\_ds.trait\_names():  
 del self.new\_ds[0x008103f]  
 self.new\_ds[0x0020000d].value = self.new\_ds[0x0020000d].value  
 self.new\_ds[0x020000E].value = generate\_uid()  
 self.ds[0x0200011].value = 2  
 self.ds[0x0200013].value = 1  
 self.ds[0x0080018].value = generate\_uid()  
 self.ds.add\_new(tag=0x0020003, VR='UI', value=self.ds[0x0080018].value)  
 self.new\_ds.save\_as('new\_DICOM.dcm')  
  
 def keyboard\_func(self, my\_key, x, y):  
 key = unicode(my\_key, errors='ignore')  
 if key == 's':  
 self.save\_to\_file()  
  
def init\_window(width, height):  
 glutInitWindowSize(width, height)  
 glutInitWindowPosition((glutGet(GLUT\_SCREEN\_WIDTH) - width) // 2, (glutGet(GLUT\_SCREEN\_HEIGHT) - height) // 2)  
 glutCreateWindow('KP\_4')  
  
def main():  
 glutInit()  
 glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB)  
 file = Image(path\_file)  
 init\_window(file.width, file.height)  
 file.init()  
 glutDisplayFunc(file.display)  
 glutKeyboardFunc(file.keyboard\_func)  
 glutMouseFunc(file.mouse\_func)  
 glutMainLoop()  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()

Результат

Після натискання клавіщі ‘s’ створюється новий DICOM файл.



**Контрольні запитання**

1. *Який порядок кодування даних пікселів в площину зображення.*

Поряждок кодування пікселів в площину зображення відбувається зліва на право та зверху вниз:

| ***1*** | ***2*** | ***3*** |
| --- | --- | --- |
| ***4*** | ***5*** | ***6*** |
| ***7*** | ***8*** | ***9*** |

1. *Пояснити поняття Pixel Cell?*

Pixel Cell – це контейнер, що зберігає значення вибірки пікселів та додаткових бітів, якщо це необхадно. Додаткові біти можуть бути використані для накладання площини та встановлення певних меж для розміщення пікселів. Розмір піксельної комірки визначається за допомогою розміщених бітів (0028, 0100) та перевищує або дорівнює збереженим бітам (0028, 0101).

1. *Що таке оверлейні дані, як відповідно до стандарту DICOM зберігається інформація щодо оверлейних даних зображення?*

В атрибуті Overlay Data (60XX, 0050) зберігається накладання як 1-но бутове зображення, стандарт DICOM дозволяє виконувати два конкретні типи такого накладання (графіка та рентабельність інвестицій). Набір таких данних може мати до 16 окремих площин переплати.

Порядок пікселів, які кодуються для кожного накладання, знаходяться зліва направо та зверху вниз.

1. *Перелічити елементи DICOM, які визначають структуру пікселів зображення: їх призначення?*

* Bits Allocated (0028, 0100) – визначає розмір кожної Pixel Cell;
* Bits Stored (0028, 0101) – визначає загальну кількість виділених бітів , що будуть використані для представлення Pixel Sample Value;
* High Bit (0028, 0102) – визначає місце розміщення біту високого порядку.

1. *Призначення елементів Rescale Intercept, Rescale Slope?*

Rescale Intercept (0028,1052) та Rescale Slope (0028,1053) – це теги, які використовуються для визначення лінійного претворення пікселів зображення, що зберігаються на диску, до їхнього представлення в пам’яті комп’ютера.

1. *Навести та пояснити формулу, відповідно до якої відбувається представлення піксельних даних в памяті для відображення?*

– де це значення даних пікселів для відображення, – нахил масштабу (коефіцієнт масштабування Rescale Slope (0028, 1053)), – значення даних піскелів, що зберігаютсья у DICOM файлі, – перехоплення масштабу (коефіціент зсуву Rescale Intercept (0028,1052)).

1. *У яких випадках застосовується лінійне масштабування?*

Лінійне масштабування використовуюється у вападках, коли значення пікселів мають дуже виликий діапазон значення, воно допомагає уникати помилок квантування та зберігати значення з меншою кількістю бітів.

1. *Пояснити, чому значення Rescale Intercept, Rescale Slope можуть бути різними для різних зображень однієї серії?*

Зображення однієї серії можуть захоплювати різні частини органів, відповідно зображення де немає присутніх патологій може мати мешні значення показників відносно зображення з патологією. Тому для таких зображення показнити Rescale Intercept (0028,1052) та Rescale Slope (0028,1053) будуть мати різні значення незалежно від серії.

1. *Що таке відповідно до стандарту похідне зображення, як його можна отримати?*

За стандартом DICOM похідне зображення – це зображення, піксельні данні якого боли отримані шляхом перетворення піксельних даних одного або декількох зображень. Коли операція над данними зображення вимагає створення нового, то таке зображення і є похідним.

1. *Як програмно задати значення унікального ідентифікатору UID, що відповідає стандарту DICOM?*

Існує вбудована функція generate\_uid(), яка генерує унікальний UID.