

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Розробка програмного забезпечення для перегляду
медичних радіологічних зображень стандарту DICOM

Виконав: студент IV курсу, групи СНЗ-41
спеціальності 122 Комп'ютерні науки

(шифр і назва спеціальності)

(підпис)

Гай В.О.

(прізвище та ініціали)

Керівник

(підпис)

Мацюк Г.Р.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Шимчук Г.В.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Боднарчук І.О.

(прізвище та ініціали)

Рецензент

(підпис)

(прізвище та ініціали)

Тернопіль - 2024

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
Боднарчук І.О.
(підпис) (прізвище та ініціали)
«__» _____ 2024 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня Бакалавр
(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки
(шифр і назва спеціальності)

Студенту Гаю Валентину Олександровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка програмного забезпечення для перегляду
медичних радіологічних зображень стандарту DICOM

Керівник роботи Мацюк Галина Ростиславівна, к.соц. ком., доц.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «29» 04 2024 року № 4/7-473

2. Термін подання студентом завершеної роботи 10.06.2024 р.

3. Вихідні дані до роботи наукові літературні джерела

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ

1. Аналіз предметної області.

2. Проектування та реалізація програмного продукту

3. Опис роботи програмного продукту

4. Безпека життєдіяльності, основи охорони праці

Висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

1. Титулка. 2. Актуальність. 3. Мета, задачі дослідження. 4. Основні характеристики дослідженого ПЗ. 5. Формування вимог до ПЗ, що буде створено.

6. Засоби програмної розробки 7. Діаграма класів розробленого ПЗ.

8. Діаграма модулів. 9. Діаграма класів із демонстрацією відношень.

10. Діаграма життєвого циклу розробки ПЗ. 11, 12. Скріншоти роботи створеного ПЗ.

13. Висновки. Основні результати проведеного дослідження

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи хорони праці			

7. Дата видачі завдання _____ 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	29.04 – 30.04.24	<i>Виконано</i>
2.	Підбір джерел про програмні засоби перегляду медичних радіологічних зображень DICOM	01.05 – 05.05.24	<i>Виконано</i>
3.	Опрацювання джерел про програмні засоби перегляду медичних радіологічних зображень DICOM	06.05 – 10.05.24	<i>Виконано</i>
4.	Виконання дослідження щодо розробки програмного забезпечення для перегляду медичних радіологічних зображень стандарту DICOM	11.05 – 15.05.24	<i>Виконано</i>
5.	Розроблення програмного коду	16.05 – 19.05.24	<i>Виконано</i>
6.	Оформлення розділу «Аналіз предметної області»	20.05 – 23.05.24	<i>Виконано</i>
7.	Оформлення розділу «Проектування та реалізація» програмного продукту»	24.05 – 27.05.24	<i>Виконано</i>
8.	Оформлення розділу «Опис роботи програмного продукту»	28.05 – 30.05.24	<i>Виконано</i>
9.	Виконання завдання до підрозділу «Безпека життєдіяльності, основи хорони праці»	31.05 – 03.06.24	<i>Виконано</i>
10.	Оформлення кваліфікаційної роботи	20.05 – 03.06.24	<i>Виконано</i>
11.	Нормоконтроль	01.06 – 05.06.24	<i>Виконано</i>
12.	Перевірка на плагіат	04.06 – 07.06.24	<i>Виконано</i>
13.	Попередній захист кваліфікаційної роботи	07.06 – 09.06.24	<i>Виконано</i>
14.	Захист кваліфікаційної роботи	11.06.24	

Студент

_____ (підпис)

Гай В.О.

_____ (прізвище та ініціали)

Керівник роботи

_____ (підпис)

Мацюк Г.Р.

_____ (прізвище та ініціали)

АНОТАЦІЯ

Розробка програмного забезпечення для перегляду медичних радіологічних зображень стандарту DICOM // Гай Валентин Олександрович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем та програмної інженерії, кафедра комп'ютерних наук, група СНз-41 // Тернопіль, 2024 // С. – 52, рис. – 21, табл. – 9, слайдів – 13, бібліогр. – 19.

Ключові слова: DICOM, воксельна графіка, медичне зображення, тривимірна візуалізація

Кваліфікаційна робота присвячена створенню відкритого кросплатформного програмного забезпечення для візуалізації медичних DICOM –зображень.

Проведено аналіз існуючих програмних засобів (з відкритим кодом та пропрієтарних) для перегляду медичних DICOM –знімків. Сформовано вимоги до програмного продукту, наведено характеристики стандарту DICOM.

Спроектовано спеціалізований програмний продукт, вибрано метод реалізації, програмні середовища та інструменти для його реалізації. Наведено модульну архітектуру продукту. Докладно описано базові та допоміжні модулі із відображенням функцій та методів, які вони реалізують. Побудована діаграма життєвого циклу процесу розробки програмного продукту.

Наведено опис роботи розробленого програмного забезпечення, починаючи від початку роботи з ним, налаштування різних параметрів візуалізації, аж до побудова тривимірної реконструкції зображення.

Результатом роботи програми є сформована 3D-модель радіологічного зображення.

ANNOTATION

Development of software for viewing medical radiological images of the DICOM standard // Hai Valentyn // Ternopil Ivan Pul'uj National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Computer Science // Ternopil, 2024 // P. - 48, Fig. - 52, Table - 9, Slide - 13, References - 19.

Keywords: DICOM, voxel graphics, medical imaging, three-dimensional visualization

Thesis deals with the creation of open cross-platform software for visualization of medical DICOM images.

An analysis of existing software tools (open source and proprietary) for viewing medical DICOM images was carried out. The requirements for the software product are formed, the characteristics of the DICOM standard are given.

A specialized software product has been designed, an implementation method, software environments and tools for its implementation have been selected. The modular architecture of the product is presented. Basic and auxiliary modules are described in detail, showing the functions and methods they implement. A life cycle diagram of the software product development process is constructed.

A description of the work of the developed software is given, starting from the beginning of working with it, setting various visualization parameters, up to the construction of a three-dimensional reconstruction of the image.

The result of the program is a formed 3D model of the radiological image.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ СКОРОЧЕНЬ І ТЕРМІНІВ

DICOM (Digital Imaging and Communications in Medicine) – стандарт для комунікації та управління медичною візуальною інформацією і суміжних даних

ISO (International Standards Organization) – Міжнародна організація зі стандартизації

NEMA (National Electrical Manufacturers Association) – Національна асоціація виробників електронного обладнання

OpenGL (Open Graphics Library) – специфікація, що визначає незалежний від мови програмування крос-платформовий програмний інтерфейс для написання застосунків, що використовують 2D та 3D комп'ютерну графіку.

OSI (Open System Interconnection) – базова еталонна модель взаємодії відкритих систем

PACS (Picture Archiving and Communication System) – технологія, що використовується в медичній візуалізації для зберігання та зручного доступу до зображень

Воксельна графіка – це метод візуалізації, що використовує тривимірні пікселі (вокселі) для створення тривимірних зображень

ГІК – графічного інтерфейсу користувача

ПЗ – програмне забезпечення

ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	10
1.1 Аналіз існуючих програмних продуктів.....	10
1.2 Формування вимог до ПЗ, що буде створено.....	12
1.3 Опис стандарту DICOM	13
РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ ПРОГРАМНОГО ПРОДУКТУ	19
2.1 Вибір методу реалізації	19
2.2 Вибір мови програмування	19
2.3 Вибір бібліотеки для розробки інтерфейсу користувача	20
2.4 Проектування архітектури ПЗ	21
2.5 Опис базових модулів програми	23
2.5.1 Клас головного вікна програми.....	25
2.5.2 Клас модуля читання DICOM файлів	27
2.5.3 Клас побудови воксельної моделі	29
2.5.4 Клас тривимірної реконструкції.....	30
2.6 Опис допоміжних модулів програми	30
2.6.1 Модуль зберігання даних DICOM.....	30
2.6.2 Модуль перехідної таблиці кольорів	31
2.6.3 Модуль налаштування програми.....	32
2.7 Життєвий цикл процесу розробки.....	32
РОЗДІЛ 3. ОПИС РОБОТИ ПРОГРАМНОГО ПРОДУКТУ	34
3.1 Початок роботи із програмою.....	34
3.2 Налаштування параметрів.....	37
3.3 Побудова тривимірної реконструкції зображення	39
РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ	41
4.1 Навчання працюючих і інструктажі з охорони праці.....	41
4.2 Санітарно-гігієнічні вимоги до умов праці.	43
ВИСНОВКИ.....	46

ПЕРЕЛІК ДЖЕРЕЛ 47

ДОДАТКИ

ВСТУП

Актуальність теми. Сьогодні медична візуалізація, окрім власне X-випромінювання, також використовує методи діагностики високочастотних хвиль, магнітних полів та інших випромінювань [1]. Зараз для формування, збереження, передавання та візуалізації медичних фото та документів пацієнтів, котрі обстежуються, використовується стандарт DICOM, який підтримується більшістю виробників медичного обладнання та ПЗ.

DICOM дає можливість здійснювати об'єднання медичного устаткування від різноманітних фірм, в тому числі сканери, сервери, автоматизовані робочі місця та принтери у цілісну систему. DICOM протокол застосовує TCP / IP для передачі медичної інформації від медичного радіологічного устаткування в PACS -систему і для зв'язку між такими системами [2].

В даний час медичні установи України прагнуть використовувати передові технології для діагностування та подальшого лікування різних захворювань. При аналізі ринку ПЗ для медичних радіологічних досліджень не було знайдено програм з відкритим вихідним кодом, які мають можливість тривимірної візуалізації та відповідають усім сучасним вимогам ПЗ. Спираючись на результати проведеного аналізу, було прийнято рішення про створення ПЗ з відкритим кодом, який матиме низку переваг платного ПЗ. Однією з основних переваг програмних продуктів з відкритим кодом є здатність кастомізації під потреби кожної конкретної медичної організації.

Мета роботи – розробка інструменту перегляду медичних зображень стандарту DICOM.

Для досягнення мети виділено ряд завдань:

- вивчити існуючі програмні продукти, провести аналіз функціональності та визначити функції, які можуть бути запозичені в програмному продукті, що розробляється;
- сформулювати вимоги до програмного продукту, що розробляється;
- на основі отриманих вимог, потрібно спроектувати модель ПЗ з урахуванням можливостей обраної архітектури. При проектуванні архітектури

- слід врахувати можливість подальшого розширення програмного середовища;
- вибрати засоби розробки ПЗ та реалізувати програмний продукт, який задовольнятиме заданим у попередніх пунктах вимогам.
 - виконати тестування програмного продукту.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Аналіз існуючих програмних продуктів

Сьогодні існує декілька спеціалізованих програмних засобів для перегляду медичних DICOM -фото. Вони можуть відрізнятися за доступним користувачем функціоналу і можуть бути як вільно розповсюджуваними, так і пропрієтарними. Для вирішення завдання формування переліку вимог до середовища програмування, що розробляється, серед інших було обрано пропрієтарне ПЗ Radiant 2.2, Multivox 5 та Inobites. Нижче буде наведено окремі їх характеристики.

Radiant - інструмент для перегляду фото DICOM-PACS. Програму було розроблено для найбільш ефективного застосування ресурсів. ПЗ здатне функціонувати із багатопроцесорними та багатоядерними системами із значним гігабайтним об'ємом пам'яті, проте водночас може успішно працювати і на старих комп'ютерах із малим об'ємом пам'яті.

Multivox 5 - розробка, що вільно розповсюджується, для роботи з X-знімками. Має багатий набір інструментів для роботи із зображеннями та інтуїтивно-зрозуміле управління.

Inobites – візуалізатор медичних даних, отриманих з різного обладнання, розгортається на діагностичних робочих станціях та інтегрується з PACS. Програмний продукт виводить можливості діагностики на інший рівень, недосяжний з використанням плівкових знімків або інших твердих носіїв, і дозволяє ефективно і своєчасно детектувати відповідні патології, проводити прогнозування їх розвитку, а також планувати їх ефективне усунення. Має кросплатформну реалізацію.

Кожна з згаданих вище програм має безліч специфікованих можливостей, які виконані для підвищення зручності та швидкості роботи з DICOM зображеннями.

В табл. 1.1 наведені основні критерії ПЗ для візуалізації радіологічних зображень (як з відкритим кодом, так і пропрієтарних).

Таблиця 1.1 – Можливості дослідженого ПЗ

Критерій / Програма	MicroDicom	MultiVox	Agnosco - Free DICOM	Radiant	Amira	Inobitec
Платформа	W	W	W	W	W	W, L
Підтримка DICOM та інтеграція PACS	+	+	+	+	+	+
Висока продуктивність для великих наборів даних	+	+	+	+	+	+
Двовимірна візуалізація	+	+	+	+	+	+
Можливість зміни налаштувань зображення (яскравість/ контрастність, інверсія інтенсивності та ін.)	-	+	-	+	+	+
Можливість масштабування зображень (збільшення/ зменшення зображень)	+	+	+	+	+	+
Тривимірна реконструкція дослідження	-	-	-	-	-	-

Тут W – Windows, L – Linux.

При розробці програмного продукту було прийнято рішення взяти всі перераховані вище можливості. Як цільову платформу було обрано кросплатформне рішення. Розглянемо перелічені можливості докладніше.

Підтримка стандарту DICOM дозволяє працювати з більшістю моделей медичного радіологічного обладнання та забезпечує єдиний формат зберігання та передачі даних [1, 2].

Висока продуктивність та багатопотоковість програмного продукту для великих наборів даних зменшує час роботи користувача при роботі з радіологічними знімками високої роздільної здатності та більш ефективного

використання автоматизованого робочого місця.

Повнофункціональна, гнучка двовимірна візуалізація дозволяє користувачеві провести більш глибоке дослідження пацієнта. І дозволяє ефективно та своєчасно виявляти відповідні патології, прогнозувати їх розвиток, а також планувати їх продуктивне усунення.

Тривимірна реконструкція дослідження дозволяє проводити віртуальні діагностичні операції та дослідження, а також дає можливість отримати нову анатомічну картину, яку не вдалося візуалізувати за допомогою лише базових зображень.

Можливість зміни налаштувань зображення. Перетворення інтенсивності дозволяють змінити яскравість або контрастність зображення, і провести інверсію інтенсивності кольору, що дає можливість змінити параметри відображення значень щільності знімка.

Можливість масштабування зображень. Геометричні перетворення зображень дозволяє збільшити або зменшити розміри зображення, що дозволяє більш детально вивчити долі зображень, що цікавлять користувача [4].

1.2 Формування вимог до ПЗ, що буде створено

Грунтуючись на аналізі існуючих програмних продуктів, були сформульовані кінцеві вимоги до розробки:

- кросплатформність. Дозволить використовувати програмний продукт, що розробляється на будь-якій платформі;
- підтримка стандарту DICOM. Дасть змогу програмі працювати з більшістю сучасних медичних апаратів;
- висока продуктивність великих наборів даних та багатопоточність. Архітектура програмного продукту повинна враховувати роботу з великим набором даних;
- можливість візуалізації двовимірних зображень;
- можливість перетворення інтенсивності зображень та ручного встановлення кольору для заданої густини знімка;

- можливість масштабування зображень. Геометричні перетворення зображення для зміни його масштабування;
- побудова тривимірної реконструкції дослідження та її візуалізація.

1.3 Опис стандарту DICOM

Є промисловим стандартом для створення, опрацювання та передавання радіологічних знімків та інших даних медичного характеру між комп'ютерами. Він ґрунтується на стандарті обстежених пацієнтів. DICOM ґрунтується на стандарті OSI, розробленим ISO [3].

Версія DICOM 1.0 була розроблена та випущена у 1985 році Американським Коледжем Радіології спільно з NEMA. У 1993 році вийшла версія DICOM 3.0. До цієї версії було додано нові службові класи, а також підтримку локальних мереж та стандартизовано Conformance Statement . На даний момент версія DICOM 3.0 є останньою версією стандарту, однак вона постійно оновлюється, що говорить про її сучасність.

Стандарт DICOM досяг широкого поширення, як засіб для медичної візуалізації, однак, він має недоліки. Оскільки DICOM є стандартом для візуалізації медичної інформації хворих, він може використовуватися до створення повноцінної медичної системи обліку даних.

Щоб вирішити проблему об'єднання медичної системи обліку даних та системи візуалізації PACS та HIS/RIS було розроблено та випущено міжнародні рекомендації, що дозволяють об'єднати системи, що працюють із протоколами DICOM та HL 7.

На даний момент версія стандарту DICOM 3.0 редакції 2009 описує інформацію, приведену в табл. 1.2.

Таблиця 1.2 – Дані зі стандарту DICOM 3.0

№ з/п	Параметр
1	Атрибути та демографічні дані пацієнта
2	Дані про медичне обладнання (модель та фірма), на якому проводилося обстеження
3	Дані про медичну установу, де було проведено обстеження
4	Атрибути персоналу, який проводив обстеження пацієнта
5	Дані про дослідження (вид, дата та час його проведення)
6	Умови та параметри проведення дослідження пацієнта
7	Параметри зображення або серії зображень, записаних у DICOM файл
8	Унікальні ключі ідентифікації груп даних, описаних у DICOM файл
9	Зображення, серію чи набір серій, отриманих під час обстеження пацієнта
10	Подання, в першу чергу, PDF документів у DICOM файл.
11	Подання DICOM записи на оптичні носії, включаючи DVD формат
12	DICOM Протокол для передачі/прийому по TCP/IP комп'ютерних мереж

Стандарт DICOM визначає два інформаційні рівні.

Файловий рівень (DICOM File) – є об'єктним файлом із теговим вмістом для виведення кадру або низки кадрів знімку, що супроводжує та керує інформацією як DICOM- теги [2].

Мережевий рівень, інша назва «комунікаційний» (DICOM Network Protocols) – використовується для передавання DICOM файлів та управляючих DICOM тегів по TCP/IP мереж [2].

Інформаційна модель DICOM File являє собою чотири ступені:

- пацієнт;
- дослідження;

- серія;
- зображення.

У DICOM файли можуть одночасно міститися і знімок дослідження, і додаткова інформація про досліджуваного пацієнта. За рахунок нероздільності інформації про пацієнта із зображенням дослідження досягається висока точність обробки даних, що знижує ризик виникнення помилок та виникнення різноманітних проблем.

Будь-який DICOM об'єкт містить множину даних, наприклад, ім'я пацієнта, дата проведеного дослідження, медичні параметри проведеного дослідження тощо. Крім того, будь-який DICOM об'єкт містить у собі особливий атрибут, що містить дані зображення (англ. pixel data). Отже, у DICOM файлу не існує якогось окремого заголовка, тільки безліч атрибутів, які включають, в тому числі, дані зображення. Атрибути в стандарті звуться теги (англ. tags). Будь-який з тегів володіє власним неповторним номером, котрий містить два поля з номерами групи та елемента. Для прикладу, те 0010, 0010 завжди має інформацію про прізвище, ім'я та по батькові пацієнта. При цьому будь-який тег володіє власною стандартним іменем. Ім'я тега 0010, 0010 буде Patient ' s Name . Тег 7 FFE , 0010 Pixel Data здатен містити у собі зображення чи декілька. Коли Pixel Data містить принаймні два зображення, стверджується, що файл має мультифреймове зображення (англ. multi - frame image).

Три або чотири розмірні зображення можуть міститися в одному файлі. Це може бути послідовність сканів томографа у кількох місцях, але зроблених у різний час. Файли у вигляді однофреймових зображень передають цифрові рентгенівські апарати, і навіть застарілі томографи. Файли мультифреймових зображень часто передають апарати УЗД та ангиографи. Сучасні томографи можуть передавати як однофреймові, так і мультифреймові зображення.

Вони можуть бути чорно-білі і кольорові. При цьому кольорові зображення можуть мати різне колірне кодування - RGB, YBR , Palette Color. Чорно-білі фото можуть мати різну глибину та градації сірого від одного до шістнадцяти біт. Дані фото можна стиснути стандартизованими алгоритмами стиснення: RLE , JPEG , JPEG Lossless , JPEG LS, JPEG 2000. Повний файл

можливо заархівувати із використанням алгоритму LZW, проте використання такого виду архівування у програмах та устаткуванні вкрай рідкісні.

Стандарт DICOM застосовує три схеми для кодування тегів. Власне кодування файлу відмічається визначеним тегом у ньому. Схеми формуються як комбінація двох атрибутів - представлення даних і, властиво, кодування байтового порядку.

Дані можуть задаватися у явному та неявному вигляді. Необхідно розуміти, яким чином інтерпретувати ті дані, котрі є в тегах, так як це всього лиш звичайний набір байт. Невідомо завчасно, які конкретно дані є в тегах - стрічка, число чи черговість тегів SQ. Для ясності було стандартизовано вміст кожного тегу. Кожен тег має стандартизований вид даних тегу VR (англ. Value Representation) - OW, OB, UN, SQ, UT, OF і т.д. За явного вигляду даних у тегах властиво явно вказується VR значення тегу. Відповідно за неявного задання VR значення не вказується, а візьметься із таблиці програми, яка працює з цим зображенням.

Порядок байт у DICOM файл можливий від найстаршого до самого молодшого, а запис вже стартує з найстаршого і завершується самим молодшим, та навпаки. Стандарт DICOM застосовує три із чотирьох максимальних варіантів атрибутів: Implicit little endian, Explicit little endian та Explicit big endian.

Будь-який тег містить: 2-байтовий номер групи, номера елемента 2 байти, 2-байтовий VR (тільки при явному заданні даних), об'єм тега 2 чи 4 байти (у залежності від VR). Ось окремі стандартні VR тегів: DA (англ. Date), дата; DS (англ. Decimal String), рядок, який є десятковим дробом; FL (англ. Floating Point Single), тип float; IS (англ. Integer String), цілий рядок; (англ. Unsigned Long).

Стандарт встановлює довжину імені файлів до восьми символів, розширення файлів неприпустимі. Одна з вимог, що склалися, до імен файлів - ім'я, не повинно містити в собі жодної персональної інформації. Це обумовлюється підтримкою оберненої сумісності до старших систем. До того ж, на носії, окрім власне файлів, у root-каталозі повинен бути файл dicomdir. Dicomdir файл являє собою узагальнену проіндексовану інформацію щодо усіх файлів DICOM на даному носії. Цей файл містить докладнішу інформацію щодо

будь-якого файлу, ніж ту, котру можна помістити власне в його імені.

Не розміщені на носії DICOM файли, як правило, мають тип . dcm, а на носії повинні бути файли без типу.

PACS є клієнт-серверною медичною інформаційною системою, що складається з взаємопов'язаних компонентів (табл. 1.3).

Таблиця 1.3 – Компоненти PACS

№ з/п	Компонент
1	Медичні DICOM обладнання, які є DICOM клієнтами
2	Одного чи кількох DICOM Серверів
3	Однією або декількома діагностичних DICOM станціями
4	Одного або кількох DICOM принтерів
–	Стандарт DICOM забезпечує цифровий зв'язок між DICOM Серверами та діагностичним обладнанням різних виробників, включаючи: <ul style="list-style-type: none">– рентгенологічну апаратуру;– ангіографічні комплекси;– комп'ютерні томографи;– магнітно-резонансні томографи;– ізотопні сканери;– ультразвукові сканери;– мікроскопи, ендоскопи;– робочі станції DICOM

Більшість сервісів, з яких складається DICOM, мають на увазі під собою обмін даними по мережі. Обмін файлами є тільки незначною його частиною, котра була додана до нього уподальшому.

Основне призначення DICOM- сервісу Store – передача зображень чи інакших об'єктів (як приклад можуть бути структуровані звіти) між якимись двома DICOM- пристроями.

Призначення сервісу Storage Commitment – надання підтвердження у

успішному поміщенні в сховищі даних інформації файлів, котрі відіслані на *servic store*.

Сервіс *Query/Retrieve* призначений для відшукування та доставляння повних досліджень чи окремих файлів і об'єктів на *DICOM*- пристрої, котрий є віддаленим. За допомогою цього сервісу можна здійснити пошук цікавого дослідження або об'єкта за визначеними фільтрами, а також запросити пересилання знайдених даних на локальну машину.

DICOM сервіс *Modality Worklist* - сервіс, що дозволяє апарату отримати перелік запланованих досліджень. У таких даних зберігається інформація про досліджуваних пацієнтів, що дає змогу скоротити повторний ввід аналогічної інформації, а також зменшити супутні помилки.

Додатковий до попереднього сервіс *Modality Performed Procedure Step* дає змогу модальності відправляти звіти щодо благополучного виконаного дослідження, його часу старту та завершення, щодо одержаної пацієнтом дози опромінення і т.п. Сервіс дає змогу будувати звіти про ефективність застосування ресурсів апарату.

DICOM сервіс *Printing* дозволяє надсилати знімки на принтер друку *DICOM* для одержання друкованої фото-копії, найчастіше на спеціальних плівках. Також сервіс *Printing* допомагає одержати ідентичні зображення як на різноманітних дисплеях, так і виведеній на папір копії зображень при допомозі стандартного калібрування принтерів та моніторів.

РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ ПРОГРАМНОГО ПРОДУКТУ

2.1 Вибір методу реалізації

Для реалізації програмного продукту було розглянуто три шляхи.

Перший шлях - написання модуля для програми роботи з тривимірною та інженерною графікою. Перевагою цього рішення є швидкість розробки. Недоліком є висока вартість програм і мала можливість масштабування цього рішення. Також ці програми дуже ресурсозатратні.

Другий шлях – написання програмного продукту на основі існуючого. Його перевагами є швидкість та простота розробки. Недолік - більшість цих програм є програмами із закритим умовним кодом через використання специфічних бібліотек або не відповідають сучасним вимогам до програмних продуктів.

Третій шлях – розробка нового програмного продукту. Цей варіант дозволяє реалізувати необхідний функціонал, використовувати зручну для розповсюдження ліцензію та дозволить надалі вільно модифікувати цей продукт. Також зменшити зміст розроблюваного ПЗ до оптимального рівня, скоротити надмірну функціональність, створити гнучкий і інтерфейс, що налаштовується, розробити кросплатформний застосунок, що легко розширюється.

Було прийнято рішення використовувати третій шлях, оскільки він дозволить виявити кращі сторони.

2.2 Вибір мови програмування

Мова програмування C++, є швидкою і такою, що, підтримується багатьма операційними системами. Її функціональність можна розширити використовуючи динамічні бібліотеки та вихідних кодів [6].

Ця мова була обрана для створення ПЗ. Вирішальним чинником вибору мови C++ стало наявність досвіду розробки ПЗ даною мовою та її

кросплатформеність.

2.3 Вибір бібліотеки для розробки інтерфейсу користувача

Для проектування ГІК існує кілька бібліотек: JUCE, GTK, wxWidgets, Qt.

Juce - це відкритий кросплатформовий інструментарій розробки графічного інтерфейсу для мови C++. Він підтримує Windows, Linux, OS X, iOS, Android. Juce поставляється у вигляді вихідного коду. Його розробкою займається одна людина. Перша публічна версія набору інструментів Juce з'явилася 2003 року. Цей набір функцій та класів дозволяє будувати різноманітні графічні інтерфейси, використовуючи лише необхідні модулі. Він безкоштовний для некомерційного використання. Об'єм усіх модулів пакета Juce дорівнює 17 мегабайтів [7].

GTK+ - це безкоштовний кросплатформовий інструментарій для створення ГІК. GTK+ підходить для будь-яких проектів. Він має простий у використанні API. Перша версія GTK+ з'явилася 1998 року. Його розмір близько 30 мегабайт [8].

wxWidgets - це кросплатформова open-бібліотека інструментів для написання програм. Основним застосуванням wxWidgets є побудова ГІК, проте бібліотека включає велику кількість інших функцій і використовується для створення різноманітного програмного забезпечення. Ця безкоштовна для користування бібліотека. Перша версія wxWidgets з'явилася 1992 року. Її розмір близько 45 мегабайт [9].

Qt - кросплатформний інструментарій розробки ПЗ на мові C++. Він складається з набору модулів, представлених у вигляді бібліотек та файлів, що підключаються. Загальний розмір бібліотек Qt 205 мегабайт. Перша версія Qt з'явилася 1996 року [10].

Для вирішення поставлених завдань було вибрано інструментарій Qt. Його вибір виправданий тим, що він включає багато бібліотек для роботи з OpenGL [11, 12].

2.4 Проектування архітектури ПЗ

Архітектура програми може бути монолітною або модульною. При використанні монолітної архітектури виникають складнощі з модифікацією програми та легкістю її доопрацювання. Монолітна організація програми різко програє у розширюваності модульної організації. На користь модульної архітектури виступила наявність множини функціональних елементів інтегрованого середовища розробки, що розробляється. Модульна архітектура дозволить легко розширити функціональність програмного продукту, що розробляється, покращувати готові модулі і забезпечує зрозумілий і легкочитаний код.

Модульна архітектура має на увазі під собою створення базового класу ПЗ, що розробляється, і множини класів функціональних модулів, яка представлена на рис. 2.1.

Як головний клас був обраний клас головного вікна програми. Як класи функціональних модулів були розроблені такі модулі:

- головне вікно програми. Модуль, через який буде організовано взаємодію всіх частин програмного продукту;
- модуль читання файлів DICOM. За його допомоги буде здійснюватись читання з DICOM файлів, та надання даних для інших модулів програми;
- модуль зберігання файлів DICOM. У ньому знаходяться класи для зберігання даних з попереднього модуля. А також функції рендерингу візуалізації в 2D зображень;
- модуль LUT. У ньому зберігається клас таблиці вхідно-вихідних RGB - значень, тобто. кожному вхідному RGB значенням відповідає своє вихідне RGB значення;
- модуль словника. В якому зберігається клас таблиці, в якій знаходитимуся описи та значень тегів DICOM;
- модуль 3D візуалізації. Призначений для перетворення даних радіологічного дослідження на воксельну модель;
- вікно 3D візуалізації. Модуль, який призначений для виведення

воксельної моделі, за допомогою OpenGL, отриманої за допомогою попереднього модуля.

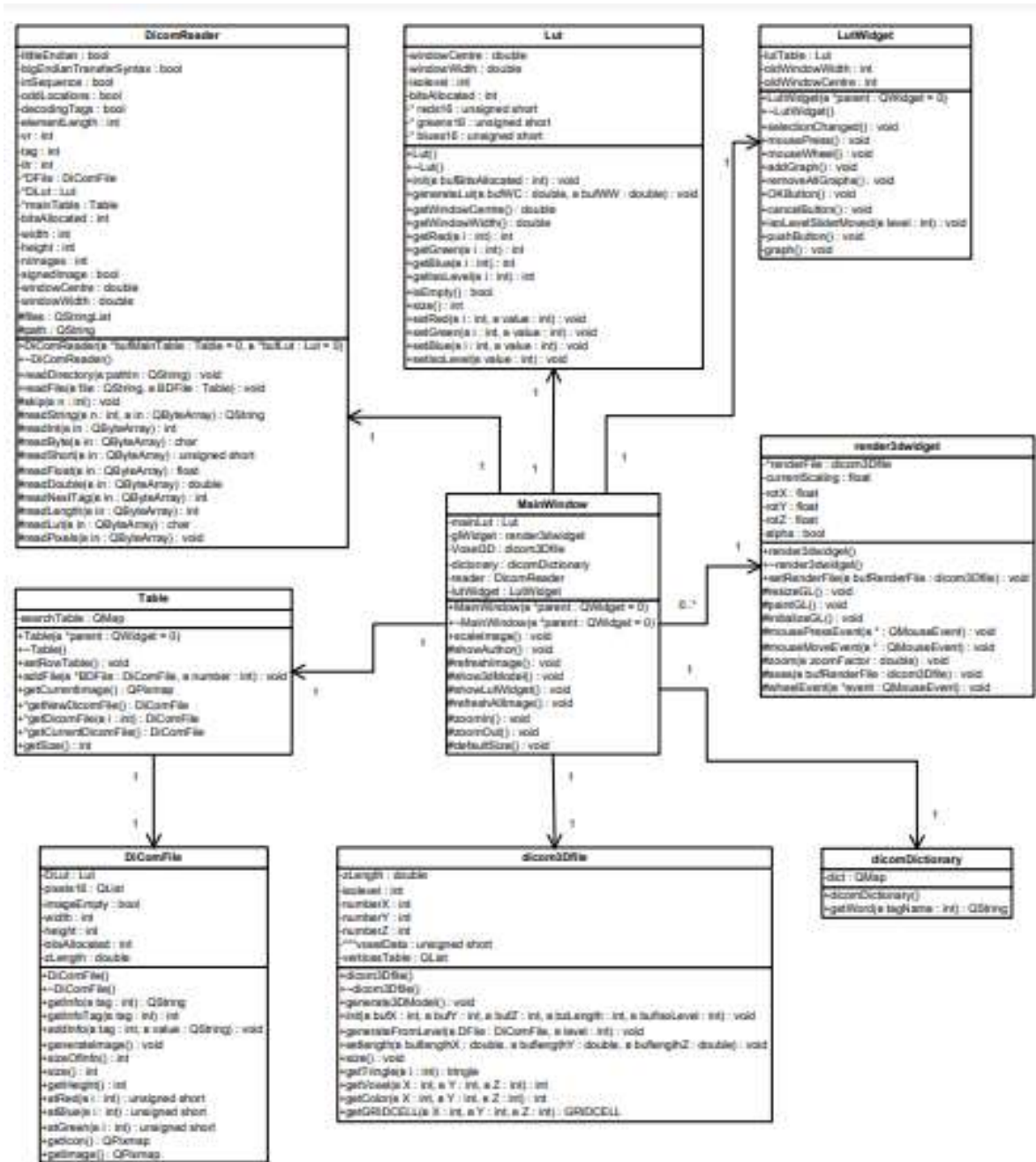


Рисунок 2.1 – Діаграма класів розроблюваного ПЗ

Крім того, цей підхід дозволяє незалежно редагувати та вдосконалювати кожен модуль окремо від усього проекту. Варто зазначити, що модулі розробляються у вигляді файлів із вихідним кодом, а не у вигляді бібліотек. Це означає, що при побудові програми необхідно всі модулі будуть побудовані заново. Цей факт не викликає складнощів, тому що всі модулі невеликого

розміру.

Зв'язок модулів продемонстровано на рис. 2.2.

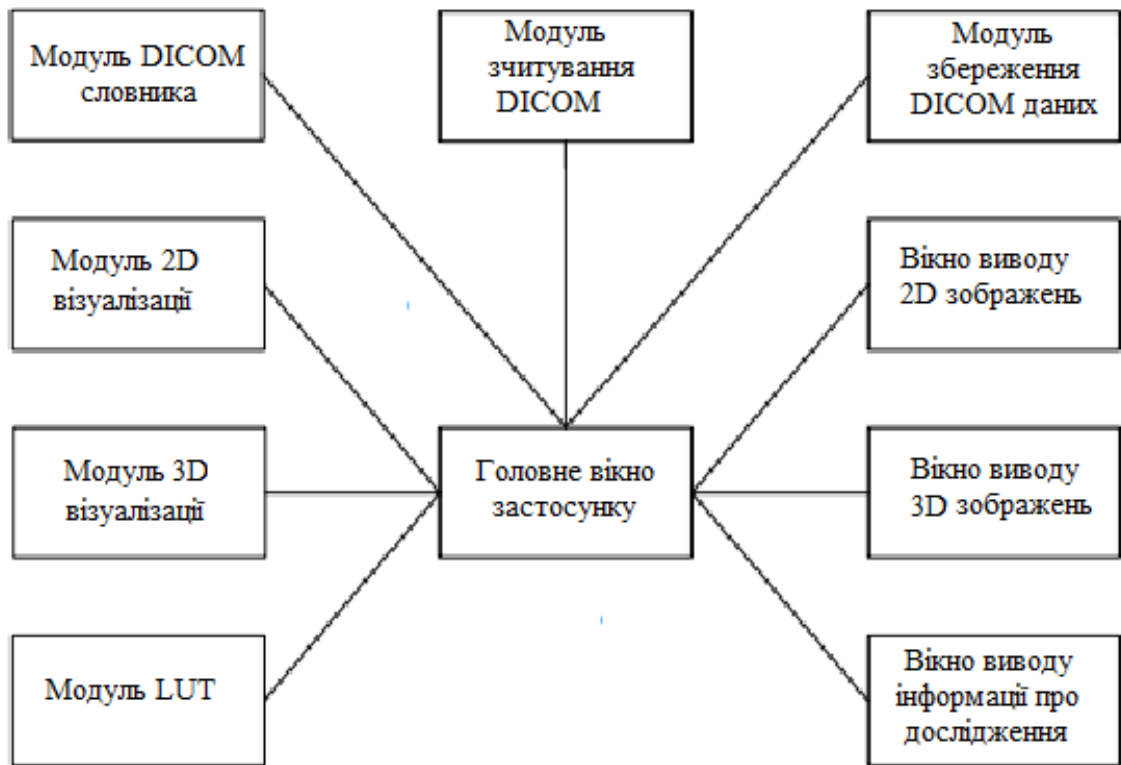


Рисунок 2.2 – Діаграма модулів

Взаємодія модулів організована через модуль головного вікна програми.

2.5 Опис базових модулів програми

Для реалізації програмного продукту проект був поділений на одинадцять частин. Кожна частина представляє окремий модуль, який містить певний функціонал програмного продукту та вирішує конкретні завдання.

На рис. 2.3 наведено діаграму класів, що демонструє відношення розроблених класів та класів бібліотеки Qt.

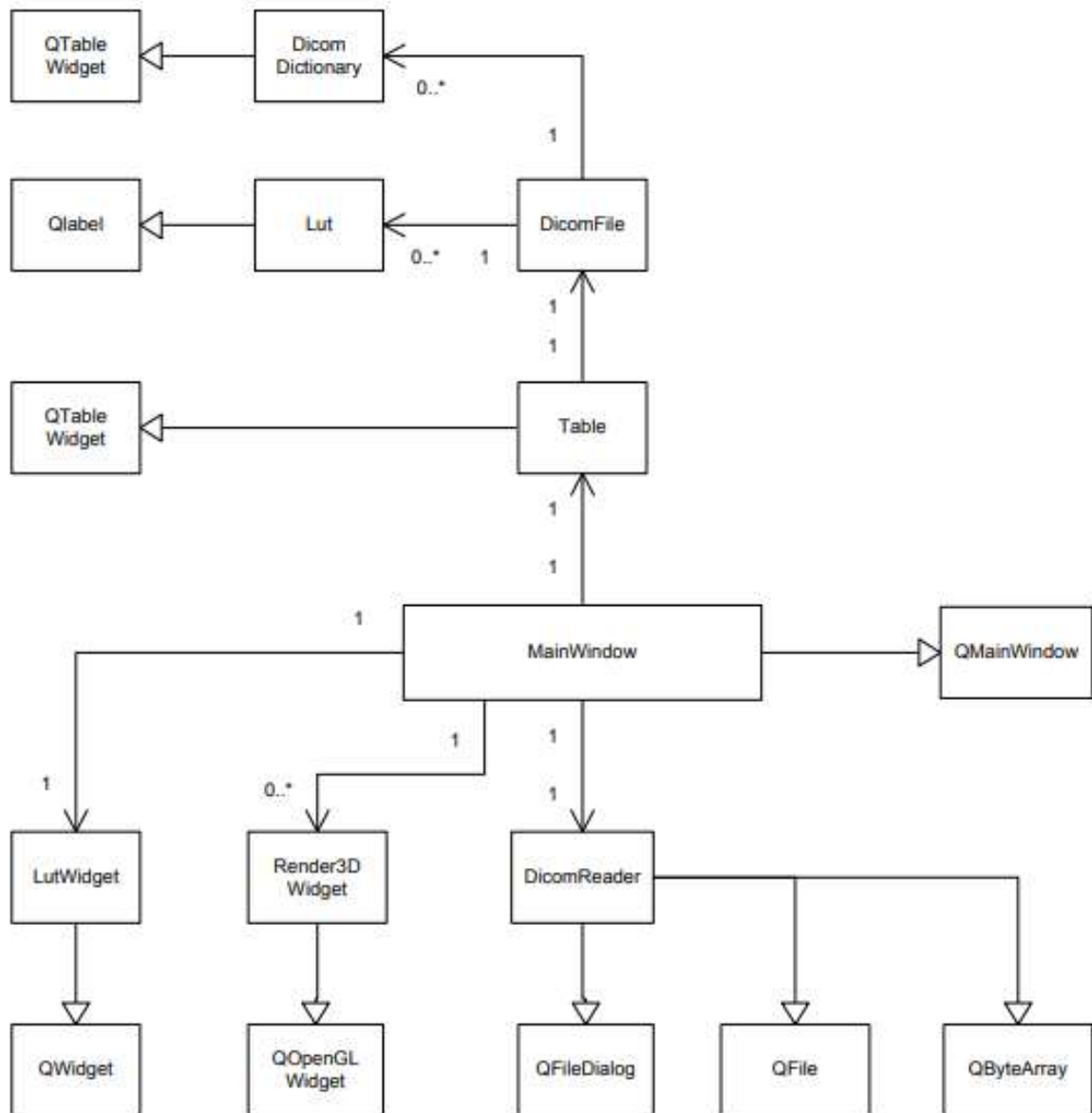


Рисунок 2.3 – Діаграма класів із демонстрацією відношень

Клас головного вікна програми успадковує QMainWindow. Є базовим класом для всіх програм, розроблених з використанням бібліотеки Qt. За допомогою цього класу здійснюється взаємодія з іншими класами програми, у тому числі цей клас надається користувачеві інтерфейс для доступу до інших класів програми.

Класи роботи з DICOM файлами служать для читання, зберігання, обробки та візуалізації даних медичних досліджень. До таких класів відносяться:

- DicomReader (читання двійкових даних DICOM файлів);
- DicomFile (зберігання даних DICOM файла);
- DicomDictionary (зберігає в собі значення та опис DICOM тегів);

- Lut (містить таблицю, необхідну для виведення зображень);
- Table (представляє список DICOM файлів);
- Dicom3DFile (створення та зберігання тривимірної моделі із серії DICOM знімків);
- LutMarchingCubes (містить заздалегідь згенеровану таблицю кубів для створення тривимірної моделі);
- LutWidget (надає інтерфейс для взаємодії зі списком налаштувань програми). Для цього LutWidget успадковує клас QWidget та використовує клас QCustomPlot;
- Render3dWidget (для відображення тривимірної моделі). Цей клас успадковує QOpenGLWidget;
- QOpenGLWidget (базовий клас для рендерингу тривимірної графіки за допомогою OpenGL).

2.5.1 Клас головного вікна програми

Клас MainWindow надає доступ до всіх інших модулів програми при допомозі панелі меню та вікон, що прикріплюються. MainWindow успадковує QMainWindow. Це треба для побудови головного вікна ПЗ. У QMainWindow встановлено панель меню, статусний рядок і сторони, у яких можна розміщувати вікна, що прикріплюються.

Для роботи з файлами та каталогами використовуються класи DicomReader, DicomFile та Table. Клас Table успадковується від класу QTableWidgetItem.

Клас Lut використовується для зберігання перехідної таблиці кольорів. А клас LutWidget, що успадковується від класу QWidget для налаштування значень, котрі зберігаються в класі Lut. Візуалізація двовимірного зображення відбувається за допомогою класу QLabel.

Для організації кросплатформних стандартних елементів ГІК були використані класи QFileDialog, QMessageBox - для вибору файлу, що відкривається, і для спливаючих повідомлень відповідно.

Для зберігання воксельної моделі та її візуалізації використовуються класи

Dicom3DFile та Render3DWidget відповідно.

До графічного інтерфейсу користувача відносяться всі підключені модулі, розроблені в рамках даної роботи. До їх переліку входять:

- MainWindow - головне вікно програми;
- LutWidget – вікно налаштування програми;
- Render3DWidget - вікно візуалізації тривимірної програми;
- Table - таблиця зі списком завантажених DICOM файлів;
- InfoTable – таблиця зі списком тегів вибраного DICOM файлу.

Список та опис функцій та об'єктів класу MainWindow, які наведені у таблиці 2.1.

Таблиця 2.1 – Перелік функцій класу MainWindow

Публічні методи класу MainWindow	
MainWindow (QWidget *parent = 0);	Функція-конструктор. У цій функції відбувається ініціалізація об'єктів, налаштування головного вікна та встановлення зв'язків між модулями програмного продукту.
void scaleImage(double factor, bool reset);	Збільшує або зменшує розмір активного зображення головного вікна.
~MainWindow();	Деструктор класу MainWindow.
Функції, що належать до розряду protected slots Як правило, це інтерфейсні функції, які викликаються у відповідь на отримання різних сигналів	
void showAuthor();	Функція відображення вікна з інформацією про розробника програми.
void refreshImage();	Функція відображення зображення та інформації про нього.
void show3dModel();	Функція відкриття вікна, де відбувається відображення тривимірної моделі.
void showLutWidget();	Функція відкриття вікна, де відбувається налаштування параметрів програми.

Продовження табл. 2.1

Функції, що належать до розряду protected slots Як правило, це інтерфейсні функції, які викликаються у відповідь на отримання різних сигналів	
<code>void refreshAllImageO;</code>	Функція оновлює зображення відповідно до поточних параметрів програми.
<code>void zoomIn();</code>	Функція збільшує поточне зображення.
<code>void zoomOut();</code>	Функція зменшує поточне зображення.
<code>void defaultSize();</code>	Функція повертає реальний розмір зображення.

Для реалізації інтерфейсу користувача були створені об'єкти `QAction` і `QPushButton`, які реагують на взаємодії користувача з різними елементами інтерфейсу і формують сигнали запуску функцій.

Частина функцій цього класу наведена у додатку А.

2.5.2 Клас модуля читання DICOM файлів

Цей клас призначений для читання двійкових даних DICOM файлів. Він називається `DicomReader` і є важливим компонентом програмного продукту. У `DicomReader` реалізовано функції читання файлів стандарту DICOM.

Він виконує читання двійкових даних:

- про пацієнта;
- про стан та характеристики обладнання;
- номер і час дослідження;
- інформацію про положення зрізу або зрізів зображень у просторі;
- версії формату даних;
- дані самого зображення.

Клас також дозволяє читати файли, як із порядком байтів від молодшого до старшого, так і в зворотному порядку.

Список та опис функцій та об'єктів класу `DicomReader` представлені у табл. 2.2.

Таблиця 2.2 – Перелік методів та параметрів класу DicomReader

Відкриті методи класу Text_Edit	
DiComReader(QObject *parent = 0);	Конструктор класу DicomReader, в якому відбувається ініціалізація та налаштування новостворених об'єктів цього класу.
DiComReader (Table *bufMainTable, Lut *bufLut, QObject *parent = 0);	Конструктор класу DicomReader, в якому відбувається ініціалізація та налаштування новостворених об'єктів цього класу. А також передаються посилання на класи Table та Lut.
void readDirectory(QString pathIn);	Метод виконує читання всіх файлів із заданої директорії.
void readFile (QString file, DiComFile * BDFile);	Метод виконує читання файлу та передає їх у вказану таблицю
Захищені методи класу Text Edit	
void skip (int n);	Метод пропускає вказану кількість байт.
QString readString(int n, QByteArray * in);	Функція зчитує вказане число байт і повертає значення типу рядок.
int readInt (QByteArray *in);	Функція прочитує значення типу int.
ushort readShort (QByteArray *in);	Функція зчитує значення типу беззнаковий short .
float readFloat (QByteArray * in);	Функція зчитує значення типу float.
double readDouble (QByteArray * in);	Функція прочитує значення типу double.
int readNextTag (QByteArray *in);	Функція повертає значення тегу даних DICOM.
int readLength (QByteArray * in);	Функція повертає кількість байт тегу даних DICOM.
char * readLut (int length, QByteArray * in);	Функція повертає значення перехідної таблиці кольорів, якщо вона вставлена у файлі.
void readPixels (QByteArray *in);	Метод зчитує дані зображення з файлу.

Клас DicomReader взаємодіє з класом головного вікна програми та з трьома класами: DicomFile, Table та LUT. Також клас містить змінні, необхідні для читання файлу. Змінна littleEndian відповідає за порядок байтів від

молодшого до старшого, а змінна `bigEndianTransferSyntax` за порядок байтів від старшого до молодшого. У змінній `itr` зберігається поточне положення в файлі, що читається. А змінні `vr` і `tag` зберігають довжину і значення тега, що зчитується.

2.5.3 Клас побудови воксельної моделі

Клас `Dicom3Dfile` призначений для створення та зберігання моделі медичних радіологічних досліджень. Також цей клас здійснює перетворення воксельної моделі на полігональну для її візуалізації за допомогою `OpenGL`.

Перетворення відбувається за допомогою алгоритму "Маршируючі куби" [13]. Відбувається покрокове проходження скалярного поля, упродовж кожної ітерації здійснює перегляд 8 сусідніх об'єктів (вершини куба, котрий є паралельним до координатних осей) та окреслює полігони, котрі потрібні для представлення фрагменти із тієї поверхні, що проходить через цей куб. Потім, на екран будуть виведені та полігони, котрі формують задану із поверхню.

У табл. 2.3 наведено перелік функцій та атрибутів класу `Dicom3Dfile`

Таблиця 2.3 – - Перелік методів та параметрів класу `Dicom3Dfile`

Відкриті методи класу <code>Dicom3Dfile</code>	
<code>Dicom3Dfile();</code>	Конструктор класу <code>Dicom3Dfile</code> .
<code>void generate3DModel();</code>	Метод виконує генерацію полігональної моделі.
<code>void init (int bufX, int bufY, int bufZ, double bzLength, int bufIsoLevel);</code>	Метод, у якому відбувається ініціалізація та налаштування новостворених об'єктів цього класу.
<code>void generateFromLevel (DiComFile * DFile, int level);</code>	Метод виконує генерацію воксельної моделі із зазначеного файлу.
Захищені методи класу <code>Dicom3Dfile</code>	
<code>int getVoxel(int X, int Y, int Z, int i);</code>	Функція повертає значення вокселя з координатами <code>x, y, z</code> .

Продовження табл. 2.3

Захищені методи класу Dicom3Dfile	
dot getColor(int X, int Y, int Z);	Функція повертає значення кольору вокселя з координатами x, y, z.
GRIDCELL getGRIDCELL(int X, int Y, int Z, int i);	Функція повертає значення кольору вокселя з координатами x, y, z.

2.5.4 Клас тривимірної реконструкції

Клас Render3DWidget відповідає за візуалізацію полігональної моделі при допомозі OpenGL. На базовому рівні OpenGL — це документ, котрий описує множину функцій з їхньою поведінкою [14]. Виробники устаткування з урахуванням OpenGL будують реалізації — бібліотеки функцій, котрі відповідають набору функцій документа. Реалізація покликана ефективно використати можливості устаткування.

Клас Render3DWidget успадковується від класу QOpenGLWidget. Клас QOpenGLWidget надає функціональні можливості для відображення OpenGL графіки, вбудовані до програми Qt.

2.6 Опис допоміжних модулів програми

2.6.1 Модуль зберігання даних DICOM

За зберігання даних у ПЗ відповідає модуль Table і DicomFile. Модуль Table є класом, що успадковує QTableWidgetItem і містить таблицю з посиланнями на дані класу DicomFile. Додавання нових файлів та очищення таблиці відбувається за допомогою методів addFile та clear відповідно.

Метод getNewDicomFile створює новий об'єкт класу DicomFile. А методи getCurrentDicomFile та getCurrentImage служать для отримання посилання на об'єкт класу DicomFile та зображення, які вибрані в графічному інтерфейсі. Модуль DicomFile призначений для зберігання даних тегів та значень густини, необхідних для відображення зображень. Також в ньому містяться значення про

кількість біт інформації необхідних для побудови зображення, а також його висота і ширина зображення (табл. 2.4).

Інші дані, які не використовуються для відображення зображення, але необхідні користувачеві програми, зберігаються в контейнері infoFile класу QList.

Таблиця 2.4 - Перелік методів та параметрів класу DicomFile

Відкриті методи класу DicomFile	
DiComFileQ;	Конструктор класу DiComFile.
QString getInfo(int tag);	Функція отримує значення тега типу QString за ітератором tag.
int getInfoTag(int tag);	Функція набуває значення тега типу int за ітератором tag.
void addInfo(int tag, QString value);	Метод виконує запис тега з його значенням.
void generateImage();	Метод виконує створення зображення.
QIcon getIcon();	Функція повертає зображення для попереднього перегляду QIcon.
QPixmap getImage();	Функція повертає зображення типу QPixmap.
int sizeOfInfo();	Функція повертає розмір контейнера з інформацією про поточний знімок.

2.6.2 Модуль перехідної таблиці кольорів

Модуль Lut необхідний для отримання кольору із значення густини тіла, що зберігається в Dicom файлі. Відповідно до стандарту DICOM, ці дані зберігаються в самому файлі або можуть бути отримані за допомогою наступного порядку дій (рис. 2.4):


```

if (x <= c - 0.5 - (w-1)/2), then y = ymin
else if (x > c - 0.5 + (w-1)/2), then y = ymax ,
else y = ((x - (c - 0.5) ) / (w-1) + 0.5) * (ymax - ymin ) + ymin,

```

Рисунок 2.4 – Порядок дій для отримання даних кольорів

Тут y_{\min} – це мінімальне значення кольору, а y_{\max} – максимальне. Значення змінних w та c знаходяться у файлі DICOM і залежать від установок апарата. Або можемо задаватися вручну у формі налаштувань програми.

2.6.3 Модуль налаштування програми

Модуль LutWidget є класом, який успадковується від класу QWidget. Цей модуль потрібний для зміни налаштувань програми. Він дозволяє змінювати значення змінних w і c , що зі свого боку викликає перебудову значень класу Lut. Також в даному модуль відбувається відображення класу Lut за допомогою бібліотеки QCustomPlot, яка потрібна для побудови графіків.

За допомогою даного модуля користувач може змінювати рівень відсікання у тривимірній побудові IsoLevel. Він необхідний для того, щоб прибрати значення з непотрібною користувачеві щільністю при побудові тривимірної реконструкції. У даному модулі також використовується змінна AlphaTest, яка відповідає, чи використовуватиметься прозорість у тривимірній реконструкції.

2.7 Життєвий цикл процесу розробки

Діаграма роботи користувача з програмою відображена на рис. 2.5.

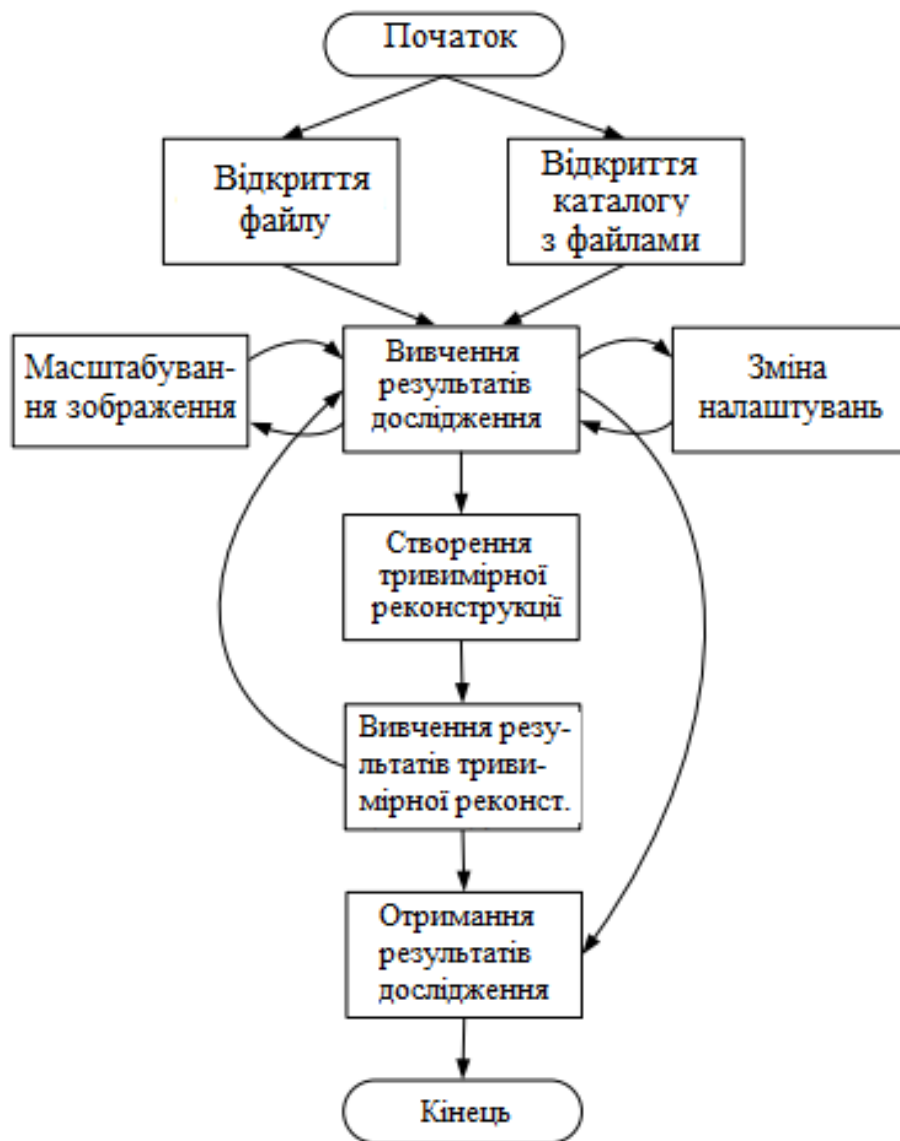


Рисунок 2.5 – Діаграма життєвого циклу розробки ПЗ

РОЗДІЛ 3. ОПИС РОБОТИ ПРОГРАМНОГО ПРОДУКТУ

3.1 Початок роботи із програмою

Після запуску програми перед користувачем відкривається головне вікно (рис. 3.1).

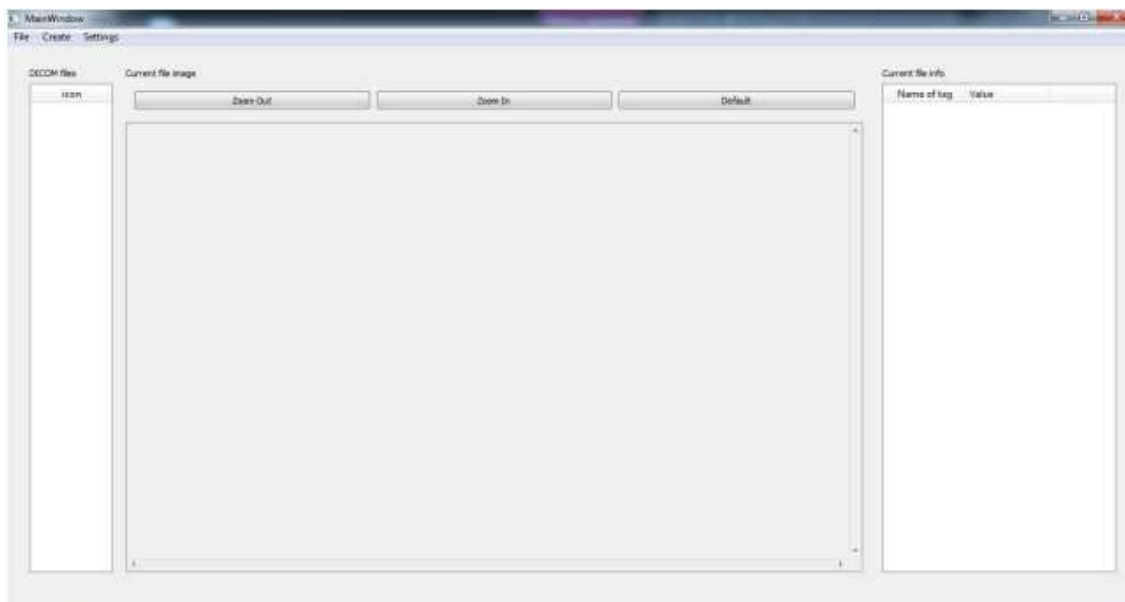


Рисунок 3.1 – Головне вікно застосунку

При відкритті файлу, перед користувачем з'являється діалогове вікно вибору файлу (рис. 3.2), де користувач вказує файл, що його цікавить.

Даний файл додається до лівої панелі, а в центрі екрана користувач бачить зображення (рис. 3.3), яке зберігається в цьому файлі:

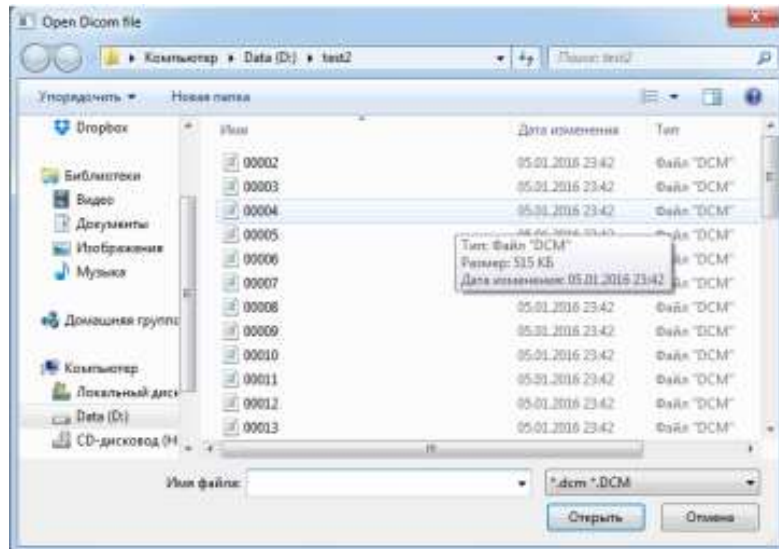


Рисунок 3.2 – Діалогове вікно вибору файлу

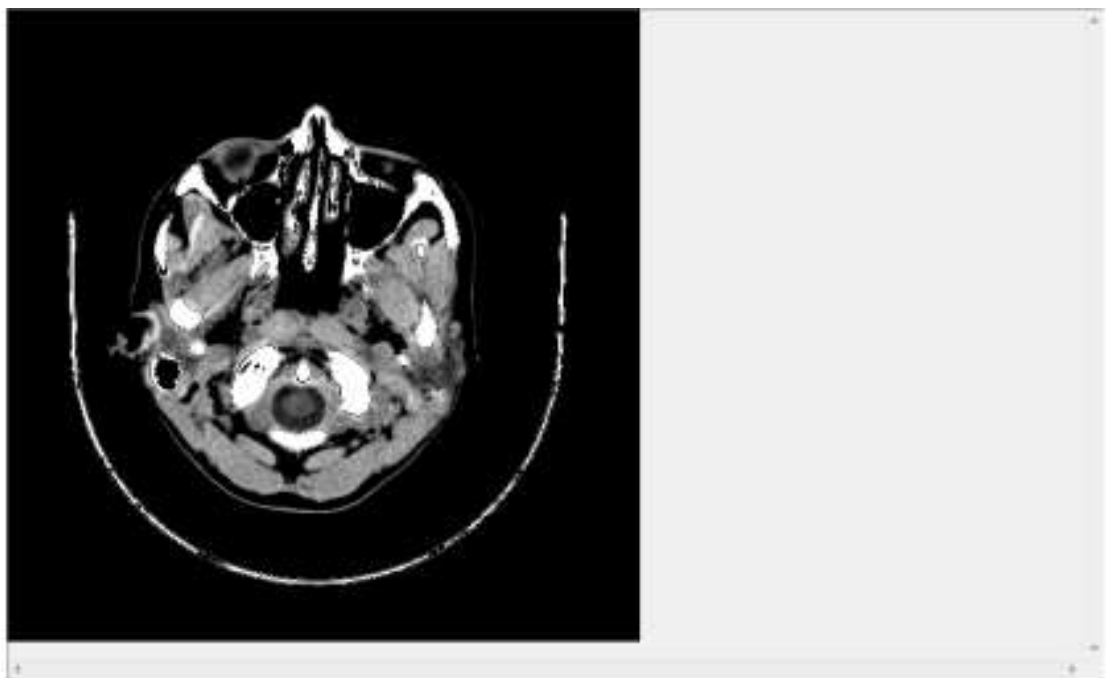


Рисунок 3.3 – Зображення файлу

Усі файли з розширенням .dcm даної директорії додаються до лівої панелі (рис. 3.4). Після відкриття файлів, користувачеві необхідно вибрати файл, що його цікавить (рис. 3.5).



Рисунок 3.4 – Панель вибору файла

Current file info		
	1	2
1	UIMedia Storag...	1.2.840.10008.5....
2	UIMedia Storag...	1.2.840.113619....
3	UITransfer Synt...	1.2.840.10008.1....
4	UIImplementati...	1.2.826.0.1.3680...
5	SHImplementat...	MAKHAON Net...
6	CSSpecific Char...	ISO_IR 100
7	CSImage Type	ORIGINAL\PRI...
8	DAInstance Cre...	20151229

Рисунок 3.5 – Панель інформації про файл

Для перегляду інформації про файл користувач має використовувати панель інформації, яка знаходиться праворуч від елемента інтерфейсу, в якому знаходиться зображення. У цій панелі знаходиться таблиця зі значення та опис тегів, що зберігаються у вибраному файлі.

При виборі іншого файлу ця інформація зміниться на значення та опис тегів іншого файлу.

3.2 Налаштування параметрів

Для масштабування зображення в програмі передбачені такі графічні елементи управління – кнопки (рис. 3.7):

- збільшення розміру зображення;
- зменшення розміру;
- відновлення стандартного розміру файлу.

Вони знаходяться вгорі над елементом інтерфейсу, де знаходиться саме зображення.

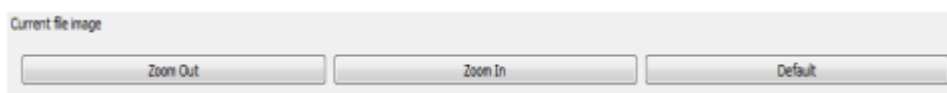


Рисунок 3.7 – Панель масштабування

Для зміни налаштувань програми користувачу необхідно відкрити вікно налаштувань програми (рис. 3.8). Для цього необхідно відкрити меню налаштувань та вибрати пункт налаштування (рис. 3.9).

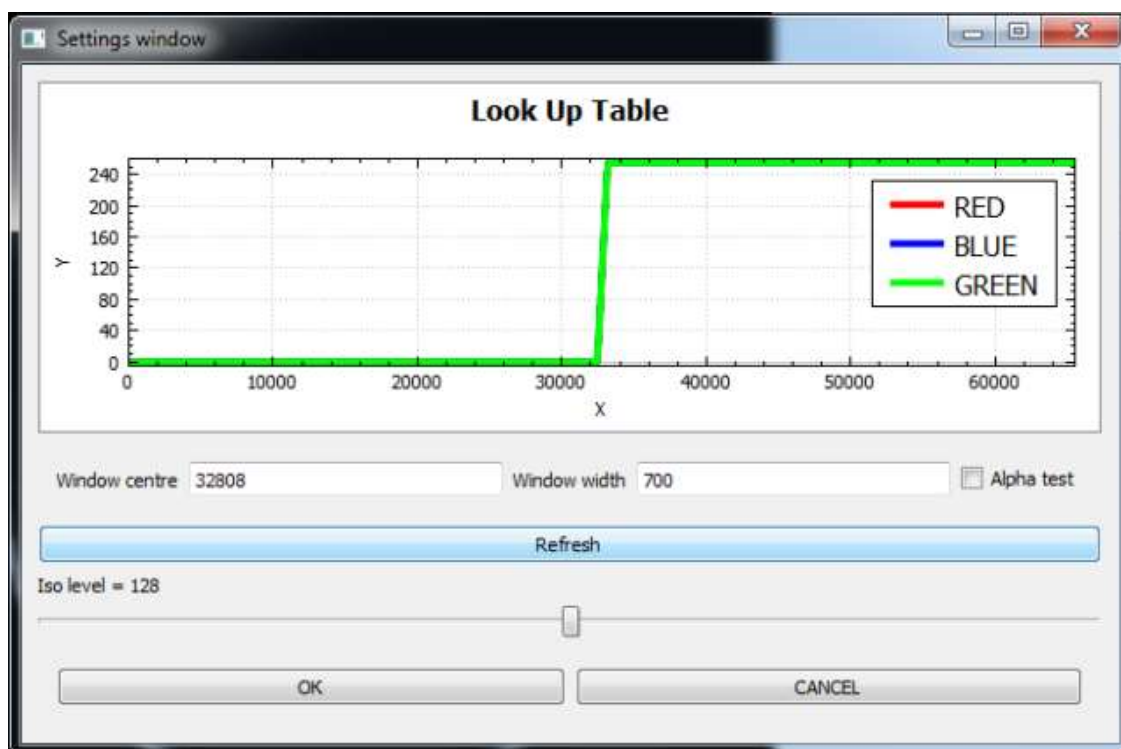


Рисунок 3.8 – Вікно налаштувань програми



Рисунок 3.9 – Меню налаштувань

У вікні параметрів програми користувач бачить графік (рис. 3.10), на якому відображається залежність значень щільності, що зберігаються у файлі від значень кольору.

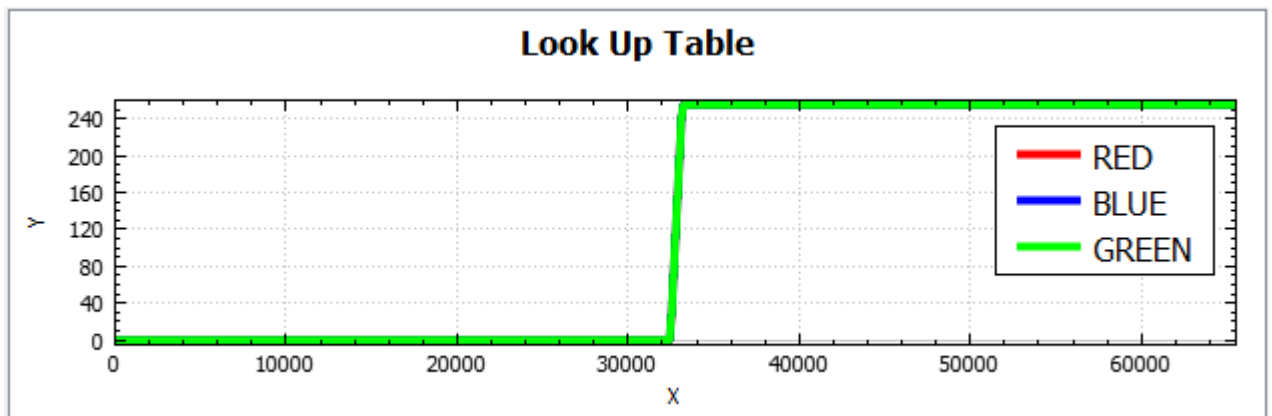


Рисунок 3.10 – Графік таблиці кольорів

Для зміни цих значень користувач може змінювати значення змінних window center і window width у полях введення, які знаходяться нижче за графік (рис. 3.11).



Рисунок 3.11 – Поля введення значень

Для перебудови графіка користувачеві необхідно натиснути кнопку оновлення графіка (рис. 3.12), яка знаходиться під полями введення змінних

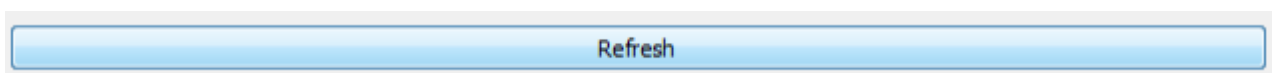


Рисунок 3.12 – Кнопка перебудови графіку

В результаті графік перебудується і згенерується нова таблиця перехідних кольорів.

Для включення режиму прозорості користувачеві потрібно буде встановити галочку елемент управління інтерфейсу відповідального за значення налаштування прозорості (рис. 3.13), при виконанні побудови тривимірної реконструкції, яка знаходиться праворуч від полів введення змінних.

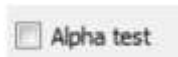


Рисунок 3.13 – Поле вибору прозорості

А для зміни рівня ізоляції необхідно змінити положення повзунка (рис. 3.14), внаслідок чого текстове поле відобразить це значення. Після завершення налаштувань користуватись повинен погодитися з вибраними налаштуваннями, натиснувши кнопку підтвердження або скасувати їх, натиснувши кнопку скасування (рис 3.15).

В результаті налаштування або набудуть чинності, або повернутися до попередніх значень.



Рисунок 3.14 – Елемент налаштування рівня ізоляції



Рисунок 3.15 – Кнопки підтвердженні і відміни налаштувань

3.3 Побудова тривимірної реконструкції зображення

Для створення тривимірної реконструкції користувачеві необхідно відкрити вікно тривимірної візуалізації.

Для цього йому необхідно вибрати меню пункт створення тривимірної

моделі (рис. 3.16).

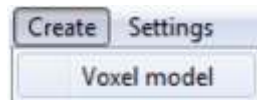


Рисунок 3.16 – Кнопки підтвердженні і відміни

У результаті перед користувачем відкриється вікно тривимірної візуалізації (рис. 3. 17), якому знаходиться модель тривимірної реконструкції дослідження, за якою користувач може побачити повноцінну форму та вигляд досліджуваних даних.

Вікно тривимірної реконструкції немає графічних елементів управління, і всі дії користувач проводить з допомогою комп'ютерної миші.

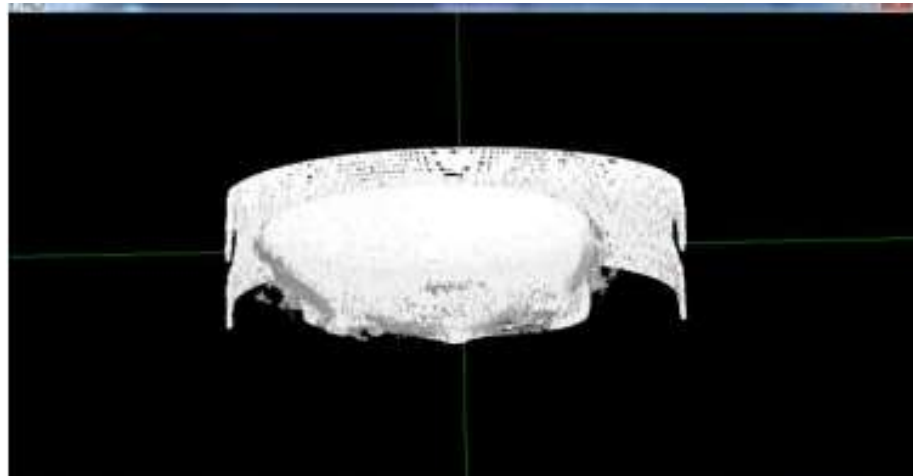


Рисунок 3.17 – Вікно тривимірної моделі

Режим масштабування тривимірної моделі здійснюється за допомогою коліщата миші. При прокручуванні назад модель віддаляється, а при прокручуванні вперед збільшується. А обертання моделі здійснюється при затисканні лівої клавіші та русі миші.

Для повороту по осях x необхідно переміщати мишу вліво або вправо в залежності від необхідного кута повороту, а для повороту по осі вгору або вниз залежно від необхідного кута повороту.

РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Навчання працюючих і інструктажі з охорони праці

Однією із складових ефективної роботи з профілактики виробничого травматизму є належна підготовка, навчання та підвищення кваліфікації працівників з питань охорони праці. Загальний порядок проведення навчання з питань охорони праці встановлений Законом України «Про охорону праці» (ст. 18. «Навчання з питань охорони праці»).

Виконання вимог Закону України «Про охорону праці» в частині проведення навчання та перевірки знань з питань охорони праці здійснюється відповідно до Типового положення про порядок проведення навчання і перевірки знань з питань охорони праці, затвердженого наказом Держкомітету України з нагляду за охороною праці 26 січня 2005 р. № 15 (далі — Типове положення).

Нагляд за дотриманням вимог Типового положення здійснюють органи державного нагляду за охороною праці, а координацію та методичний супровід — Головний навчально-методичний центр та навчальні підрозділи експертно-технічних центрів Держгірпромнагляду.

Вивчення предмета «Охорона праці» при підготовці, перепідготовці та підвищенні кваліфікації працівників, які залучаються до виконання робіт з підвищеною небезпекою, на підприємстві регламентується п. 2.3 Типового положення. На підприємствах згідно з п. 1.1 Додатку 3 Типового положення навчання та перевірку знань з питань охорони праці повинні проходити керівники, заступники керівників, головні спеціалісти, керівники основних виробничих та технічних служб, безпосередньо пов'язані з організацією безпечного ведення робіт. Крім цього, згідно з п. 5 Додатку 3, навчання та перевірку знань з питань охорони праці мають проходити керівники, спеціалісти служб охорони праці, члени комісій з перевірки знань з питань охорони праці, особи, відповідальні за технічний стан і безпечну експлуатацію об'єктів

підвищеної безпеки підприємств.

Типове положення встановлює порядок та місце проведення навчання та перевірки знань з питань охорони праці посадових осіб (п. 5.2 та п. 5.3). Посадові особи, перелік яких наведено в п. 5.2, проходять навчання у Головному навчально-методичному центрі Держнаглядохоронпраці. Перевірка знань цієї категорії посадових осіб проводиться комісією, створеною наказом Держгірпромнагляду.

Організацію навчання та перевірки знань з питань охорони праці працівників на підприємстві здійснюють працівники служби кадрів або інші спеціалісти, яким роботодавець доручив організацію цієї роботи. Навчання та перевірка знань з питань охорони праці працівників (виконавців і посадових осіб), які не залучаються до виконання робіт підвищеної безпеки, проводиться не рідше ніж один раз на три роки. Посадові особи та працівники, які виконують роботи підвищеної безпеки, проходять спеціальне навчання та перевірку знань відповідних нормативно-правових актів з охорони праці не рідше одного разу на рік.

Посадові особи малих підприємств (п. 5.4), які не мають можливості створити власні комісії з перевірки знань з питань охорони праці та провести навчання з питань охорони праці, проходять навчання та перевірку знань в навчальних закладах, які мають відповідний дозвіл.

Спеціальне навчання з питань охорони праці може проводитись безпосередньо на підприємстві або навчальним закладом, який має відповідний дозвіл. При проведенні такого навчання на підприємстві навчальні плани та програми розробляються з урахуванням конкретних видів робіт, виробничих умов і функціональних обов'язків працівників і затверджуються наказом керівника підприємства.

Періодичність інструктажів, навчання та перевірки знань з питань охорони праці залежить від видів виконуваних робіт та встановлюється Типовим положенням. Перевірка знань з питань охорони праці після проведення спеціального навчання проводиться комісією підприємства.

Якщо на підприємстві неможливо створити комісію з перевірки знань з

питань охорони праці (п. 4.4 Типового положення), перевірка знань проводиться комісією спорідненого підприємства або Теруправління Держгірпромнагляду.

Всі працівники та посадові особи підприємства, включаючи посадових осіб, відповідальних за виконання робіт підвищеної небезпеки (крім зазначених в п. 5.2 та п. 5.3 Типового положення), проходять навчання та перевірку знань з питань охорони праці на підприємстві. Типове положення не зобов'язує, але й не забороняє проводити навчання всіх виконавців та посадових осіб (особливо тих, що виконують роботи підвищеної небезпеки) в навчальних закладах. У нашій країні є багато підприємств, де таке навчання проводиться, і це має позитивні наслідки. Ті витрати, які при цьому несуть підприємства, перекриваються створенням більш безпечних умов праці і в результаті збереженням життя та здоров'я працівників.

Також в навчальних закладах проходять навчання та перевірку знань із загальних питань охорони праці всі посадові особи та фахівці, які проводять інструктажі або навчання підлеглих працівників з питань охорони праці, виконують роботи з проектування об'єктів, а також інші працівники, незалежно від того, передбачено таке навчання Типовим положенням чи ні.

4.2 Санітарно-гігієнічні вимоги до умов праці

На робочих місцях працівників, які відповідальні за експлуатацію сервісу управління механізмом авторських прав на мультимедійні файли, необхідно забезпечити дотримання вимог, затверджених Наказом Мінсоцполітики від 14.02.2018 за № 207 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями».

Приміщення для роботи з ЕОМ мають бути обладнані системами опалення, кондиціонування повітря, або припливно-витяжною вентиляцією. У приміщеннях на робочих місцях мають забезпечуватись оптимальні значення параметрів мікроклімату: температури, відносної вологості та рухливості повітря відповідно до норм та правил, а також ДБН В.2.5-67:2013 «Опалення, вентиляція та кондиціонування», затверджених наказом Мінрегіону від 25.01.2013 р. № 24.

Відповідно до санітарних норм мікроклімату виробничих приміщень ДСН 3.3.6.042-99 в офісних приміщеннях, обладнаних ЕОМ, температура повітря повинна становити 22-25°C, відносна вологість повітря – 40-60 %, швидкість руху повітря – не більше 0,1 м/с [17].

Приміщення, призначені для роботи з ЕОМ, повинні мати природне освітлення. У виробничих приміщеннях, обладнаних ЕОМ, необхідно створити належне освітлення. При експлуатації сервісу управління механізмом авторських прав на мультимедійні файли, важливим, з точки зору охорони праці, є забезпечення достатньої величини природного та штучного освітлення, які визначені у НПАОП 0.00-7.15-18 [18]. Природне світло повинно бути бічним, зорієнтованим, як правило, на північ чи північний схід, і забезпечувати коефіцієнт природної освітленості не нижче 1,5%. При виробничій потребі дозволяється експлуатувати ЕОМ у приміщеннях без природного освітлення за узгодженням з органами Держпромгірнагляду та органами й установами санітарно-епідеміологічної служби. Вікна приміщень повинні мати регулювальні пристрої для відчинення, а також жалюзі, штори тощо. Штучне освітлення приміщення з робочими місцями, обладнаними відеотерміналами ЕОМ загального та персонального користування, має бути всеосяжним і рівномірним. У випадку, коли переважають роботи з документами, допускається комбіноване освітлення (додатково до загального освітлення встановлюється світильники місцевого освітлення). Світильники розміщуються збоку від робочих місць (переважно ліворуч), або локально над робочим місцем (при розташуванні відеотерміналів ЕОМ за периметром приміщення). Як джерело світла при штучному освітленні застосовуються, як правило, люмінесцентні лампи. У світильниках місцевого освітлення допускається застосування ламп розжарювання. Рівень освітленості на робочому місці має становити 300-500 лк. При використанні комбінованого освітлення не допускається відблисків на поверхні екрана та збільшення освітлення екрана вище 300 лк.

Орієнтація вікон повинна бути на північ або північний схід, вікна повинні мати жалюзі, які можна регулювати, або штори; не дозволяється розміщувати кабінети обчислювальної техніки у підвальних приміщеннях будинків; кабінети,

обладнані комп'ютерною технікою, в навчальних закладах повинні розміщуватись в окремих приміщеннях з природним освітленням та організованим обміном повітря; стіни, стеля і підлога та обладнання кабінетів комп'ютерної техніки повинні мати покриття із матеріалів з матовою фактурою з коефіцієнтом відбиття: стін — 40- 50 %, стелі — 70 - 80 %, підлоги — 20-30 %, предметів обладнання — 40-60 % (робочого столу — 40-50 %, корпуса дисплею та клавіатури — 30-50 %, стелажів — 40-60 %); поверхня підлоги повинна мати антистатичне покриття та бути зручною для вологого прибирання; забороняється використовувати для оздоблення інтер'єру приміщень комп'ютерних кабінетів полімерні матеріали (дерев'яно-стружкові плити, шпалери, що придатні для миття, плівкові та рулонні синтетичні матеріали, шаровий паперовий пластик та ін.), що виділяють у повітря шкідливі хімічні речовини, які перевищують гранично допустимі концентрації; вміст шкідливих хімічних речовин в повітрі дошкільних та учбових приміщень з комп'ютерною технікою не повинен перевищувати середньодобові концентрації [17].

Організація робочого місця фахівця із експлуатації сервісу повинна забезпечувати відповідність усіх елементів робочого місця та їх розташування ергономічним вимогам ДСТУ 8604:2015 «Дизайн і ергономіка. Робоче місце для виконання робіт у положенні сидячи. Загальні ергономічні вимоги».

Відстань від екрана до ока фахівців, які працюють за комп'ютером визначається згідно з вимогами ДСанПіН 3.3.2.007-98.

Рівень шуму не повинний перевищувати: на місцях, де працюють програмісти та оператори ЕОМ, 55 дБА, у лабораторіях, де складаються алгоритми та ведеться робота з документацією – 60 дБА, у машинному залі – 65 дБА, у приміщеннях, де розміщені гучні агрегати обчислювальних машин – 75 дБА.

ВИСНОВКИ

У процесі виконання кваліфікаційної роботи було вивчено основні характеристики різних програм для візуалізації та перегляду даних медичних радіологічних знімків. По завершенню аналізу програмних продуктів, були сформувані вимоги до ПЗ, що розробляється.

Для створення ПЗ був обраний найкращий і оптимальний для даного випадку набір інструментів та рішень, до якого увійшли: мова програмування C++ та кросплатформова бібліотека графічного інтерфейсу Qt, а також бібліотека для побудови графіків та візуалізації даних QCustomPlot [19]. Це дало змогу у повному обсязі реалізувати сформульовані вимоги до ПЗ, що розробляється.

Під час розробки даного ПЗ, було вивчено стандарт DICOM, структурою організації тегів, а також вивчено алгоритми перетворення воксельної графіки в полігональну. Крім того, були вивчені раніше незнайомі можливості бібліотек Qt. Ґрунтуючись на здобутих знаннях, мною була спроектована архітектура програмного продукту, що розробляється, і виконана його реалізація у вигляді програми.

Цей програмний продукт має всі переваги інших подібних програм, при цьому будучи кросплатформовою програмою з відкритим вихідним кодом.

ПЕРЕЛІК ДЖЕРЕЛ

1. Ковальський О. В. Радіологія. Променева терапія. Променева діагностика : підручник для студ. вищ. мед. навч. заклад. / О. В. Ковальський, Д. С. Мечев, В. П. Данилевич. — Вінниця : Нова Книга, 2013. — 512 с. : іл.
2. Стандарт DICOM [Електронний ресурс] – Режим доступу: [https://learn.ztu.edu.ua/pluginfile.php/233171/mod_resource/content/0 М3Л2%20-%20Стандарт%20DICOM.pdf](https://learn.ztu.edu.ua/pluginfile.php/233171/mod_resource/content/0_M3Л2%20-%20Стандарт%20DICOM.pdf) (Дата звернення: 28.02.2024).
3. Офіційний сайт організації, яка розробляє стандарт DICOM. [Електронний ресурс] – Режим доступу: <http://dicom.nema.org/> (Дата звернення: 28.02.2024).
4. Feeman, Timothy G. The Mathematics of Medical Imaging: A Beginner's Guide / Feeman, Timothy G. – New York: Springer, 2010 – 141 p.
5. Marc Levoy. Display of Surfaces from Volume Data / Marc Levoy. – North Carolina: CG&A, 1988 – 10 p.
6. Васильєв О. В. Програмування на C++ в прикладах і задачах : Навч. посіб. – Київ : Видавництво Ліра-К, 2017. – 382 с.
7. Опис бібліотеки Juce // Juce CrossPlatform C++ Library. [Електронний ресурс] – Режим доступу: <http://www.juce.com/> (Дата звернення: 03.03.2024).
8. Опис бібліотеки GTK+ // The GTK+ Project. [Електронний ресурс] – Режим доступу: <http://www.gtk.org/> (Дата звернення: 04.03.2024).
9. Опис бібліотеки wxWidgets // wxWidgets: Cross-Platform GUI Library. [Електронний ресурс] – Режим доступу: <https://www.wxwidgets.org/> (Дата звернення: 04.03.2024).
10. Документація Qt 5.4. [Електронний ресурс] – Режим доступу: <http://doc.qt.io/qt-5/classes.html> (Дата звернення: 05.03.2024).
11. Гайдуков С. OpenGL Профессиональное программирование трехмерной графики на C++ / С. Гайдуков – Санкт-Петербург: БХВ-Петербург, 2012 – 722 с.
12. Офіційний сайт специфікації OpenGL. Official suite of OpenGL. [Електронний ресурс] – Режим доступу: <https://www.opengl.org/> (Дата звернення: 05.03.2024).

20.04.2024).

13. William E. Lorensen, Harvey E. Cline: Marching Cubes: A high resolution 3D surface construction algorithm / W. E. Lorensen, H. E. Cline – New York: Computer Graphics, 1987 – 167 p.

14. Основні документи, які описують специфікацію OpenGL. [Електронний ресурс] – Режим доступу: <http://opengl.org/> (Дата звернення: 20.04.2024).

15. Вступ до програмування мовою C++. Організація даних / Т. О. Карнаух, Ю. В. Коваль, М. В. Потієнко, А. Б. Ставровський. – К.: ВПЦ "Київський університет", 2015. – 276 с.

16. Шилдт Г. Полный справочник по C++ / Г. Шилдт – Москва: Вильямс, 2015 – 800 с.

17. Зеркалов Д.В. Безпека життєдіяльності та основи охорони праці. Навчальний посібник. К.: «Основа». 2016. – 267 с.

18. Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями. [Електронний ресурс] - Режим доступу: <https://zakon.rada.gov.ua/laws/show/z0508-18> (Дата звернення: 01.06.2024).

19. QCustomPlot. [Електронний ресурс] - Режим доступу <https://www.qcustomplot.com/> Дата звернення: 24.04.2024)

ДОДАТКИ

Фрагмент лістингу розробленого програмного забезпечення

Функція створення рішення.

```
void DiComReader::readFile(QString fileName, DiComFile *BDFile)
{
DFile = BDFile; itr = 0;
QFile file(fileName);
    if(file.open(QIODevice::ReadOnly))
{
    QByteArray *in;
    QString format = "DCIM"; skip(128);
    in = new QByteArray(file.readAll());
    if(format !=readString(4,in));
    //
    decodingTags = true;
    while(itr<in->size()){ while
        (decodingTags){ int tag =
            readNextTag(in);
if (inSequence)
{
DFile->addInfo(tag, NULL);
continue;
//qDebug()<<itr;
QString s;
switch (tag)
{
case TRAN SFER_SYNT AX_UID: s = readString(elementLength,
    in);
DFile->addInfo(tag, s);
if (s.indexOf("1.2.4") > -1 || s.indexOf("1.2.5") > -1)
{
file.close();
```

```
//return false;
}
if (s.indexOf(" 1.2.840.10008.1.2.2") >= 0)
    bigEndianTransferSyntax = true; break;

case MODALITY:
{
QString modality = readString(elementLength, in);
DFile->addInfo(tag, modality);
break;
}

case NUMBER_OF_FRAMES:
{ s = readString(elementLength, in); DFile->addInfo(tag,
    s); double frames = s.toDouble(); if (frames > 1.0)
    nImages = (int)frames;
.....
.....
```

Функція отримання зображення із файлу.

```
void DiComReader::readPixels(QByteArray *in)
{
if (samplesPerPixel == 1 && bitsAllocated == 8)
{
if(!DFile->pixels8.isEmpty())
DFile->pixels8.clear();
//DFile->pixels8 = new QList<char>;

int numPixels = width * height; char * buf = new char
    [numPixels]; itr = offset;
for(int i=0;i<numPixels;i++) buf[i] = readByte(in);
for (int i = 0; i < numPixels; ++i)
{
int pixVal = (int) (buf [i] * rescaleSlope +
    rescaleIntercept); if (photointerpretation ==
    "MONOCHROME1") pixVal = max8 - pixVal;
DFile->pixels8.append((char) (pixelRepresentation == 1 ?
    pixVal
: (pixVal - min8)));
}
}
if (samplesPerPixel == 1 && bitsAllocated == 16)
{
if (!DFile->pixels16.isEmpty())
DFile->pixels 16.clear(); if (!pixels16Int.isEmpty())
    pixels16Int.clear();
//pixels16 = новий QList<ushort>;
//pixels16Int = новий QList<int>;
int numPixels = width * height;
char* bufByte = new char[numPixels*2];
char* signedData = new char[2];
itr = offset;
```

```

for(int i=0;i<numPixels*2;i++) bufByte[i] = readByte(in);
    ushort unsignedS; int i, i1, pixVal; char b0, b1;

for (i = 0; i < numPixels; ++i)
{
i1 = i * 2;
b0 = bufByte[i1];
b1 = bufByte[i1 + 1];
unsignedS = ushort((b1 << 8) + b0);

if (pixelRepresentation == 0)
{
pixVal = (int) (unsignedS * rescaleSlope +
    rescaleIntercept); if (photointerpretation == M
    MONOCHROME1 M ) pixVal = max16 - pixVal;
}
else
{
//signedData[0] = b0;
//signedData[1] = b1;
short sVal = (short)unsignedS-min16;
pixVal = (int) (sVal * rescaleSlope + rescaleIntercept);
    if (photointerpretation == "MONOCHROME1") pixVal = max16
    - pixVal;
}
pixels 16Int.append(pixVal);
}
int minPixVal = pixels16Int.first(); for(int i = 0;i
    <pixels16Int.size();i++)
if(minPixVal < pixels16Int.at(i))minPixVal =
    pixels16Int.at(i); signedImage = false; if (minPixVal <
    0) signedImage = true;
.....

```