

# 2020 Spring OOP Assignment Report

과제 번호 : ASSN5

학번 : 20190439

이름 : 오승훈

Povis ID : sho0927

## 명예서약 (Honor Code)

나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다.

I completed this programming task without the improper help of others.

프로그램을 하다 보면 결정해야 할 세부 사항이 많은데, 이러한 세부 사항을 처리한 방법과 이유를 보고서에 쓰십시오.

독창적인 아이디어와 추가 기능은 보너스 점수를 받을 수 있으므로, 보고서에 명확히 기재하십시오.

제출 기한이 지나면 LMS를 통하여 과제를 제출할 수 없습니다. 제출 기한이 지난 후에는 담당 조교에게 email로 보내야 합니다. 늦게 제출된 과제에 대한 감점은 담당 조교에게 email이 도착한 시간을 기준으로 합니다.

문제가 여러 개인 경우, 각 문제별로 정리해서 작성합니다.

각 문항별 설명은 편의를 위한 것으로, 삭제하고 제출한다.

## 1. 프로그램 개요

### a. ASSN5

- A. ASSN5은 qt를 이용하여 Event-driven programming의 기초를 익히고, GUI 기반의 코딩을 해보는 것을 목표로 한다.
- B. 이번 ASSN5의 경우 휴대폰 어플에 기본적인 Watch / Stopwatch / Timer 제작을 목표로 한다.
- C. QT는 GUI 기반이기 때문에 widget을 기본 단위로 사용하는 것을 특징으로 하고 있다. 그렇기 때문에 UI의 centralWidget 내부에 Tabwidget을 넣고 그 안에 다른 Widget을 넣어서 구현을 하였습니다.
- D. ASSN5의 이름은 ASSN5\_3으로 되어있습니다!! 그래서 ASSN5\_3.exe 파일로 지금 현재 저장되어 있습니다!

## 2. 프로그램의 구조 및 알고리즘

a. Mainwindow 클래스

Mainwindow private 변수	<pre> private slots:     void myfunction();     void startClicked();     void stopClicked();     void resetClicked();     void update_time();      void timer_startClicked();     void timer_stopClicked();     void timer_resetClicked();     void timer_hour_upClicked();     void timer_min_upClicked();     void timer_sec_upClicked();     void timer_hour_downClicked();     void timer_min_downClicked();     void timer_sec_downClicked();     void timer_update();     void on_tabWidget_tabBarClicked(int index);  private:     Ui::MainWindow *ui;     QTimer *timer_watch;     QTimer *timer_stopwatch;     QTimer *timer_timer;     QMediaPlayer* player;      int minute;     int second;     int mmsec;      int timer_hour;     int timer_minute;     int timer_second; </pre>
Mainwindow public 함수	<pre> <b>MainWindow</b>(QWidget *parent = nullptr); ~<b>MainWindow</b>(); void alram_on(); </pre>

A. 이번 클래스는 UI에 모든 기반을 두고 있는 코딩이었습니다. 그렇기 때문에 UI에

필요한 것들을 추가해주기 위한 함수들과 변수들만 존재하고 나머지는 UI에 해당하는 값을 주기 위한 함수들이라고 생각해주시면 될 것 같습니다.

- B. 먼저 private slot 들에 대해서 설명을 드리자면 myfunction()의 경우 처음 만들어 보는 것을 기반으로 했기 때문에 이름이 myfunction이라고 되어 있고, 이 함수는 정확히는 시계의 현재 시간을 표시해주기 위한 slot과 연결되어 있는 함수입니다.
- C. Start/stop/resetClicked 함수의 경우 stopwatch의 버튼과 연결되어 있는 함수로 이 버튼을 누를 시 스탑워치가 시작되거나 멈추거나 리셋이 되게 해주는 함수로 구현이 되어있습니다.
- D. Update\_time의 경우 stopwatch의 시간을 실시간으로 바꿔주기 위한 함수로 0.01 초에 한 번씩 신호가 가게 되어 시간을 0.01씩 늘려주는 함수입니다.
- E. Timer\_start/ stop / reset의 경우 위의 stopwatch의 start/stop/resetClicked 함수랑 동일한 작동을 하는 함수입니다.
- F. Timer\_hour\_up / down 등과 같은 함수들은 timer의 버튼에 각각의 up / down 버튼과 연결이 되어있습니다. 그러므로 버튼이 클릭 될 시 신호가 보내져 1을 올려 주거나 1을 감소시켜주는 함수를 기반으로 하고 있고, 만약 hour를 예시로 보자면 up 함수에서는 24가 되면 0이 되도록 down 함수에서는 -1이 되면 23이 되도록 만드는 함수로 기반이 되어있습니다.
- G. Timer\_update의 경우 stopwatch의 시간을 바꿔주는 update\_time과 동일한 함수입니다.
- H. On\_tabWidget\_tabBarClicked의 경우 tabWidget을 centralwidget 안에 있는 widget으로 넣고, 그 tabWidget 내부에 모든 Watch / Stopwatch / Timer를 만들어 두었는데 그 기능들로 바꾸게 해주는 기능을 하게 됩니다. 그러므로 Watch tab이 눌리지면 watch 기능이 나오는식으로 동작이 되어있습니다. 그리고 이 것의 경우 UI 파일에 기반을 두었기 때문에 UI 파일에서 어떻게 구현이 되어있는지 확인을 할 수 있습니다. 또한 Watch 모드가 기본으로 설정이 되어야 했기 때문에 UI 탭에서 watch 모드가 기본으로 될 수 있도록 watch 모드의 tab이 나오는 UI로 설정을 해 두었습니다.
- I. alarm\_on 함수의 경우 timer에서 00:00:00이 되면 신호가 가서 알람이 울리도록 해주는 함수로 기반을 두고 있습니다.
- J. Private 에서 minute / second / mmsec 는 stopwatch의 시간을 표현해주기 위한 int 값들입니다. 또한 timer\_hour / minute / second는 말 그대로 timer의 시간 값들을 표현해주기 위한 int 값들을 저장해주고 있는 값입니다.

- K. timer\_watch / timer\_stopwatch / timer\_timer의 경우 각각의 시간 슬롯을 담당하고 있습니다. Timer\_watch의 경우 1초당 한 번 신호가 가서 watch 모드의 시간이 현재 시간으로 설정이 되도록 해주고, timer\_stopwatch의 경우 0.01초에 한 번 신호가 가서 stopwatch의 시간이 설정이 되도록 해줍니다. Timer\_timer의 경우 1초에 한 번 신호가 가서 1초에 1씩 깎이도록 해줍니다.
  - L. QMediaPlayer player 의 경우 resource 파일에 들어가 있는 alarm.wav 파일을 틀어주기 위한 변수입니다.
- b. Main cpp
- A. Main 함수의 경우 다른 것을 만들 필요가 없었습니다. UI를 기반으로 하는 QT 코딩을 진행하였고, 그렇기 때문에 main 함수에서 해줄 것은 이벤트가 들어오도록 return 값으로 기다리는 것 밖에 없어서 원래 함수를 그대로 사용하였습니다.
- c. UI 파일
- A. UI 파일에서는 centralWidget 안에 tabWidget을 넣은후 tabWidget도 3개의 tab을 만들어서 첫 번째 tab에는 watch 모드가 세팅이 되도록 label을 하나 넣어 두었고, stopwatch의 경우 label 한 개와 3개의 버튼으로 구성이 되어있습니다. 그리고 마지막으로 timer의 경우 label 한 개와 9개의 버튼으로 구성이 되어있습니다. 이들은 모두 mainwindow 클래스 내부 메소드로 연결이 되도록 해렸습니다.

### 3. 토론 및 개선

- 이번 코딩에서는 다른 것보다 QT라는 것을 이해하기가 쉽지는 않았던 것 같습니다. QT라는 것이 사실 새로운 언어를 배우는 느낌이 나도록 만드는데 있었고, 그렇기 때문에 코드가 이렇게 된다는 걸 이해하기가 쉽지 않았던 것 같습니다. 또한 QT에서의 return a.exec() 라는 함수가 어떻게 동작하는 것인지를 이해하기 위해서는 디버깅을 돌려보면서 어떻게 함수가 돌아가는지를 이해하는 방법이 제일 최선이었던 것 같습니다.
- 그리고 widget 추가를 위해서 UI를 기반으로 하면 됐던 것인데 UI 내부에서 tabWidget이라는 것이 있다는 것을 늦게 알게 되어 watch / stopwatch / timer 구현은 쉽게 하였는데 이것들을 어떻게 다른 widget 끼리 연결을 시키지? 라는 고민을 많이 했던 것 같습니다. 그러다보니 처음에는 UI 파일을 사용하지 않고 만드는 코딩을 하였다가 실패하여 한 번 갈아 엮고 코딩을 했었습니다...
- 또한 disconnect 라는 것을 몰라서 stopwatch를 예시로 들자면 stopwatch에서 start를 누르고 stop을 누르고 다시 start를 누르면 0.01초에 한 번 신호가 가는 것은 동일하나 0.01초가 올라가는 것이 아니라 0.02초가 올라가고 또 한 번 반복하면 0.01초

에 0.03초씩 올라가는 오류를 발견하였습니다. 그러다가 알게 된 것이 disconnect라는 기능이었고, slot이 연결되어있는 상태에서 disconect를 해주지 않으면 신호가 한번이 아니라 connect된 횟수 만큼 간다는 것을 그때 이해하게 되었습니다.