

융합프로그래밍실습2_3조

2020088359 김경현, 2022045332 김시온,

2022022488 박준호, 2022088913 이재욱

로봇융합전공

DEPARTMENT OF ROBOT ENGINEERING



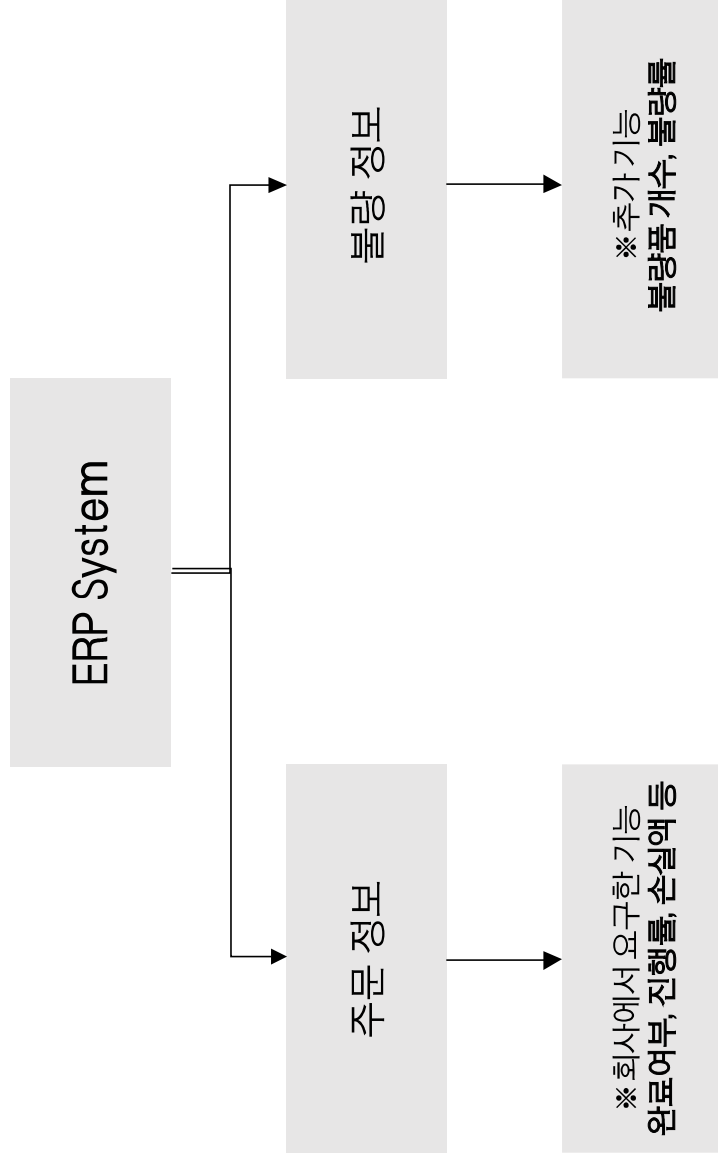
목 차

Contents

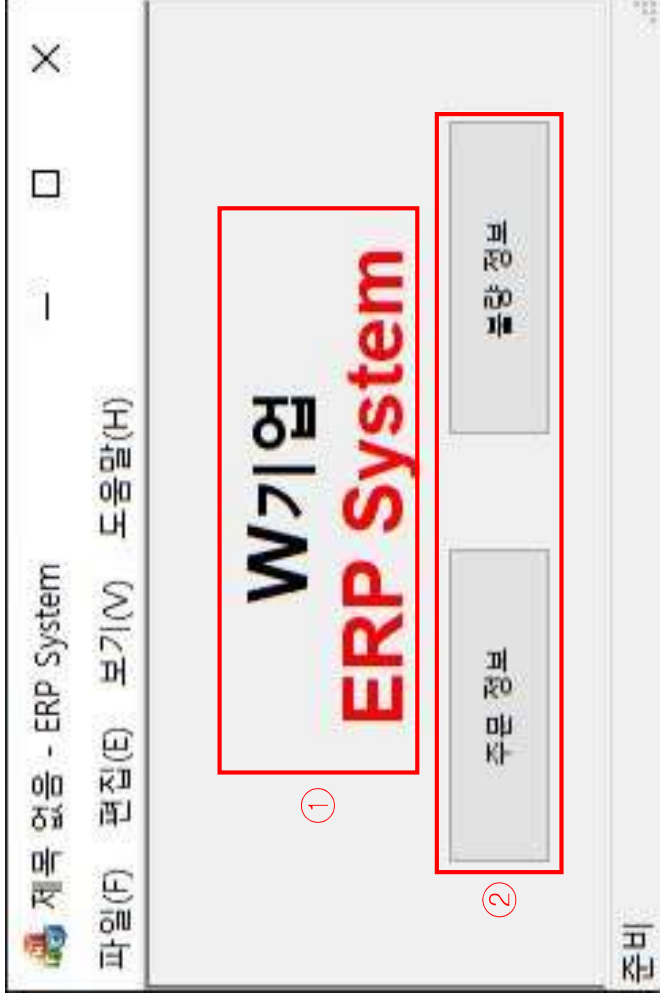
- I. 프로젝트 소개
- II. ERP System 디자인
 - 시작 Dialog & 주문 정보 Dialog & 불량 정보 Dialog
- III. ERP System 기능
 - 시작 Dialog & 주문 정보 Dialog & 불량 정보 Dialog
- IV. 코드 구현
- V. 동작 영상

프로젝트 소개

- 우리는 W기업 연구센터에 소속된 개발3팀의 개발자들
- 3주 전 회사로부터 발주 내역을 관리하기 위한 프로그램 개발을 지시 받음
- 우리는 프로그램 개발을 완료했으며 프로그램의 이름은 “ERP System”



ERP System 디자인 - 시작 Dialog



① : 프로그램명 (OnDraw)

② : 각 정보 Dialog 생성 버튼 (Button)

The Engine of Korea
공장과 스톱, 활력이 양 날개를 펼쳐 미래로 날아가는
한양대 ERICA

더불어
전국
합의
운동
본부

ERP System 디자인 - 주문 정보 Dialog

- ① : 검색 방식 선택 (Radio Button)
- ② : 날짜 선택 (Combo Box)
- ③ : 주문 건수 선택 (List Box : Multiple)
- ④ : 검색 버튼 (Button)
- ⑤ : 검색 결과 표시 (List Box : None)

ERP System 디자인 - 불량 정보 Dialog

불량 정보

검색

○ 불량 검색

○ 선택 검색

>

>

주요자

ID

번호

형식

검색

①

②

③

품목 1 불량품	Static	Static
품목 2 불량품	Static	Static
품목 3 불량품	Static	Static
품목 4 불량품	Static	Static
품목 5 불량품	Static	Static

- ① : 각 품목 색상 (Picture Control)
- ② : 각 품목 불량품 개수 및 불량률 (Static Text)
- ③ : 불량률 원형 그래프 (Picture Control)

로봇융합전공

The Engine of Korea
공감의 소통, 참여와 협력의 양 날개를 펼쳐 미래로 나아가는
한양대 ERICA



The Engine of Korea
공장과 소용, 용하여 세계의 양 부가를 펼쳐 미래를 밝히는
한양대 ERICA

- ① : Radio Button을 사용하여 날짜 검색 & 검색 범위 선택
- ② : List Box(결과)에 검색된 기간에 대한 주만, 월, 일, 년, 시, 분, 초를 표시

ERP System 기능 - 주문 정보 Dialog

①

날짜	일	주문자 ID
2020	01	0
2020	02	1
2020	03	2
2020	08	3
2020	09	0
2020	10	5
2020	11	4
2020	12	1
2021	01	3
2021	02	0
2021	03	6
2021	08	2
2021	09	4
2021	10	1
2021	11	0
2021	12	3
2022	01	2
2022	02	0
2022	03	3
2022	08	6
2022	09	8
2022	10	7

②

날짜	일	주문자 ID
2020	01	0
2020	02	1
2020	03	2
2020	08	3
2020	09	0
2020	10	5
2020	11	4
2020	12	1
2021	01	3
2021	02	0
2021	03	6
2021	08	2
2021	09	4
2021	10	1
2021	11	0
2021	12	3
2022	01	2
2022	02	0
2022	03	3
2022	08	6
2022	09	8
2022	10	7

③

오류

시작하는 날짜가 끝나는 날짜보다 빨라야합니다.

확인

① : 날짜 검색 선택 시 List Box(검색)와 선택 검색 Button 비활성화

② : 선택 검색 선택 시 Combo Box와 날짜 검색 Button 비활성화

③ : 날짜 검색 Button 클릭 시 시작 날짜가 끝 날짜보다 클 경우 오류 메시지 출력

Etc. 각 Radio Button 클릭 시 List Box(검색)의 선택된 커서들 & List Box(출력) 초기화

ERP System 기능 - 주문 정보 Dialog

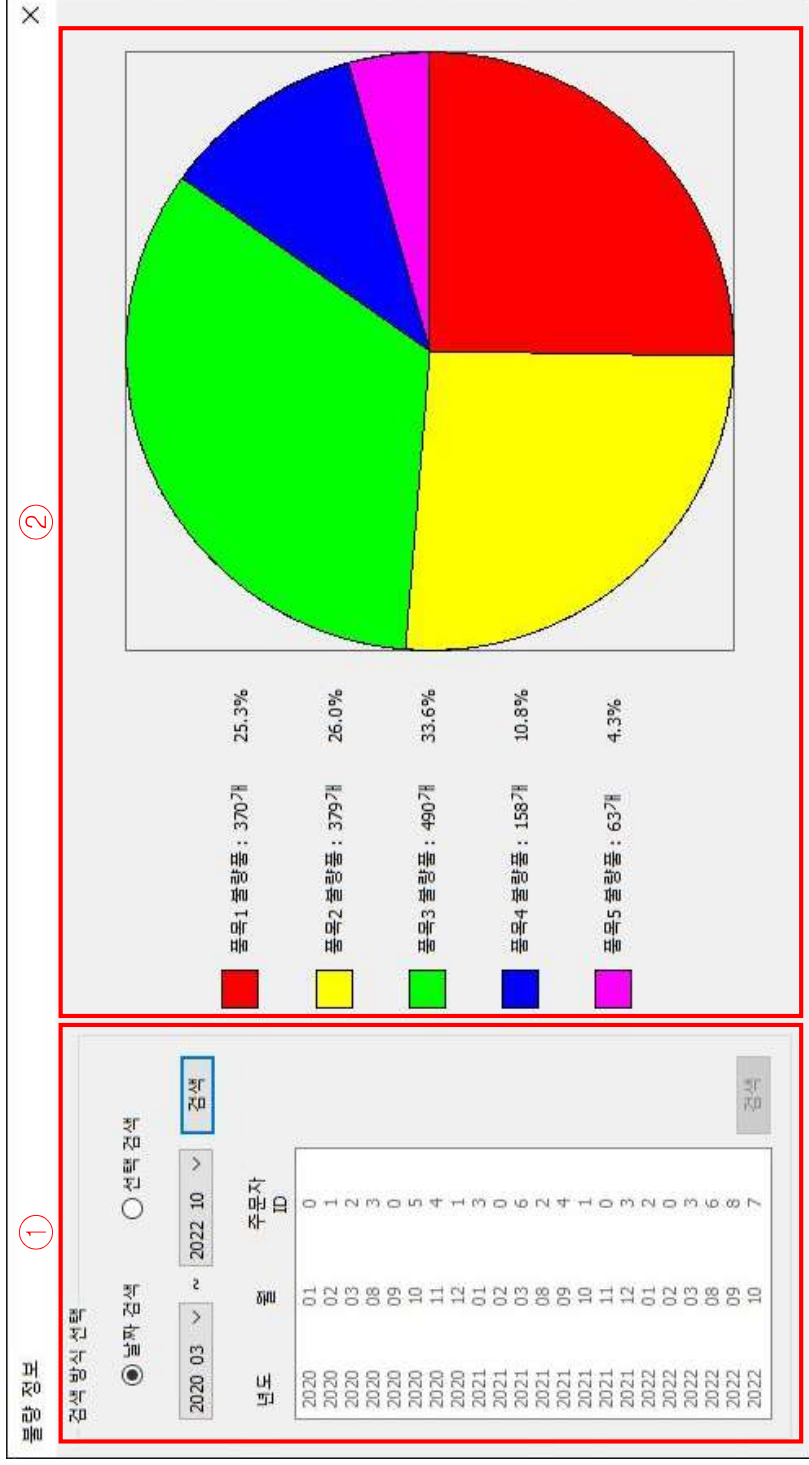
①										②				③			
주문내역										진행률				수실액			
년도	월	주문자 ID	품목1 주문내역	품목2 주문내역	품목3 주문내역	품목4 주문내역	품목5 주문내역	주문내역 합계	입력 여부	가공부 수	품질관리팀 수	품질관리팀 수	총 진행률	가공부 수	품질관리팀 수	품질관리팀 수	손실액 합계
2021	03	6	300	0	200	300	0	800	0	0	0	0	100%	0	0	0	₩ 701000
2021	08	2	300	200	500	500	100	1600	0	0	0	0	100%	0	0	0	₩ 1745000
2021	09	4	200	0	400	0	500	1100	0	0	0	0	100%	0	0	0	₩ 1028000
2021	10	1	100	200	0	0	300	600	0	0	0	0	100%	0	0	0	₩ 508000
2021	11	0	200	0	300	500	0	1000	0	0	0	0	100%	0	0	0	₩ 796000
2021	12	3	100	300	400	0	0	800	0	0	0	0	100%	0	0	0	₩ 1270000
2022	01	2	500	300	600	600	300	2300	0	0	0	0	100%	0	0	0	₩ 1027000
2022	02	0	300	0	200	200	0	700	0	0	0	0	100%	0	0	0	₩ 488000
2022	03	3	200	500	200	0	0	900	0	0	0	0	100%	0	0	0	₩ 995000
2022	08	6	300	0	500	200	0	1000	X	35	84	0	94%	35	84	0	₩ 341000
2022	09	8	0	300	0	700	600	1600	X	428	481	0	71%	428	481	0	₩ 713000
2022	10	7	500	0	600	0	800	1900	X	736	795	0	59%	736	795	0	₩ 598000

① : 주문 내역 - 년도, 월, 주문자 ID, 각 품목의 주문내역 및 주문내역 합, 완료 여부

② : 진행률 - 가공부와 품질관리팀의 남은 업무량 및 총 진행률

③ : 손실액 - 가공부와 품질관리팀에서 발견한 불량품의 손실액 및 총 손실액

ERP System 기능 - 불량 정보 Dialog



① : Radio Button을 사용하여 날짜 검색 & 선택 검색으로 구분하여 검색

② : 각 품목별 불량품 개수와 불량률을 Text와 원형 그래프로 출력

<파일 입력>

```
void CAllData::GetData()
{
    ifstream fin;
    fin.open("IC-PBLdata.txt");
    int i = 0;
    int j = 0;
    while (!fin.eof()) {
        fin >> read_val[i][j];
        j++;
        if (j == 28) {
            i++;
            j = 0;
        }
    }
}
```

<Combo Box & List Box 데이터 입력>

```
void CAllData::FillComboBox()
{
    for (int i = 0; i < 22; i++) {
        CString str;
        str.Format(_T("%d %02d"), read_val[i][0], read_val[i][1]);
        m_start.AddString(str);
        m_end.AddString(str);
    }
}

void CAllData::FillSearchListBox()
{
    m_search_list.ResetContent();
    for (int i = 0; i < 22; i++) {
        CString str;
        str.Format(_T("%d %02d %d"),
            read_val[i][0], read_val[i][1], read_val[i][2]);
        m_search_list.AddString(str);
    }
}
```

<주문 정보 검색>

```
void CallData::OnBtnClickedButton1()  
{  
    m_result_list.ResetContent();  
    int start = m_start.GetCurSel();  
    int end = m_end.GetCurSel();  
    if (start > end) {  
        MessageBox(_T("시작하는 날짜가 끝나는 날짜보다 빨라야합니다."), _T("오류"), MB_OK | MB_ICONERROR);  
    }  
    else {  
        for (int i = start; i <= end; i++) {  
            FillResultListBox(i);  
        }  
    }  
}  
  
void CallData::OnBtnClickedButton2()  
{  
    m_result_list.ResetContent();  
    int cnt = m_search_list.GetCount();  
    for (int i = 0; i < cnt; i++) {  
        if (m_search_list.GetSel(i)) {  
            FillResultListBox(i);  
        }  
    }  
}
```


$$\begin{array}{c} \wedge \\ \text{하} \\ \text{하} \\ \text{하} \\ \text{하} \\ \text{하} \\ \text{하} \\ \vee \end{array}$$
[illegible]

<불량 정보 검색>

```
void CEachData::OnBtnClickedButton1()  
{  
    int start = m_start.GetCurSel();  
    int end = m_end.GetCurSel();  
  
    sum_errr[0] = 0;  
    sum_errr[1] = 0;  
    sum_errr[2] = 0;  
    sum_errr[3] = 0;  
    sum_errr[4] = 0;  
  
    if (start > end) {  
        MessageBox(_T("시작하는 날짜가 끝나는 날짜보다 빨라야합니다."), _T("오류"), MB_OK | MB_ICONERROR);  
    }  
    else {  
        for (int i = start; i <= end; i++) {  
            SumError(i);  
            FillResultChart();  
        }  
    }  
}  
  
void CEachData::OnBtnClickedButton2()  
{  
    int cnt = m_search_list.GetCount();  
  
    sum_errr[0] = 0;  
    sum_errr[1] = 0;  
    sum_errr[2] = 0;  
    sum_errr[3] = 0;  
    sum_errr[4] = 0;  
  
    for (int i = 0; i < cnt; i++) {  
        if (m_search_list.GetSel(i)) {  
            SumError(i);  
            FillResultChart();  
        }  
    }  
}
```


<불량 정보 출력>

```
void CEachData::SumError(int i)
{
    for (int j = 0; j < 5; j++) {
        sum_err[j] += read_val[i][18+j] + read_val[i][23+j];
    }

    str.Format(_T("%d"), sum_err[0]);
    m_error_item1.SetWindowTextW(str + "개");
    str.Format(_T("%d"), sum_err[1]);
    m_error_item2.SetWindowTextW(str + "개");
    str.Format(_T("%d"), sum_err[2]);
    m_error_item3.SetWindowTextW(str + "개");
    str.Format(_T("%d"), sum_err[3]);
    m_error_item4.SetWindowTextW(str + "개");
    str.Format(_T("%d"), sum_err[4]);
    m_error_item5.SetWindowTextW(str + "개");

    total = sum_err[0] + sum_err[1] + sum_err[2] + sum_err[3] + sum_err[4];

    for (int i = 0; i < 5; i++)
    {
        per[i] = (float)(sum_err[i]) / total * 100;

        str.Format(_T("%z,1f"), per[0]);
        m_per1.SetWindowText(str + "%");
        str.Format(_T("%z,1f"), per[1]);
        m_per2.SetWindowText(str + "%");
        str.Format(_T("%z,1f"), per[2]);
        m_per3.SetWindowText(str + "%");
        str.Format(_T("%z,1f"), per[3]);
        m_per4.SetWindowText(str + "%");
        str.Format(_T("%z,1f"), per[4]);
        m_per5.SetWindowText(str + "%");
    }
}
```

<원형 그래프 출력-1>

```
void CEachData::FillResultChart()
{
    x[0] = (int)Xwidth * cos((360 * per[0] / 100) * pi / 180)) + rectview.CenterPoint().x;
    y[0] = (int)Xheight * sin((360 * per[0] / 100) * pi / 180)) + rectview.CenterPoint().y;

    x[1] = (int)Xwidth * cos((360 * (per[0] + per[1]) / 100) * pi / 180)) + rectview.CenterPoint().x;
    y[1] = (int)Xheight * sin((360 * (per[0] + per[1]) / 100) * pi / 180)) + rectview.CenterPoint().y;

    x[2] = (int)Xwidth * cos((360 * (per[0] + per[1] + per[2]) / 100) * pi / 180)) + rectview.CenterPoint().x;
    y[2] = (int)Xheight * sin((360 * (per[0] + per[1] + per[2]) / 100) * pi / 180)) + rectview.CenterPoint().y;

    x[3] = (int)Xwidth * cos((360 * (per[0] + per[1] + per[2] + per[3]) / 100) * pi / 180)) + rectview.CenterPoint().x;
    y[3] = (int)Xheight * sin((360 * (per[0] + per[1] + per[2] + per[3]) / 100) * pi / 180)) + rectview.CenterPoint().y;

    ClientDC dc(this);

    CBrush brush[5];
    brush[0].CreateSolidBrush(piecolor[0]);
    brush[1].CreateSolidBrush(piecolor[1]);
    brush[2].CreateSolidBrush(piecolor[2]);
    brush[3].CreateSolidBrush(piecolor[3]);
    brush[4].CreateSolidBrush(piecolor[4]);

    dc.SelectObject(&brush[0]);
    dc.Rectangle(&rectcolor1);
    dc.SelectObject(&brush[1]);
    dc.Rectangle(&rectcolor2);
    dc.SelectObject(&brush[2]);
    dc.Rectangle(&rectcolor3);
    dc.SelectObject(&brush[3]);
    dc.Rectangle(&rectcolor4);
    dc.SelectObject(&brush[4]);
    dc.Rectangle(&rectcolor5);
}
```

<원형 그래프 출력-2>

```

if (sum_err[0])
{
    dc.SelectObject(&brush[0]);
    dc.Ellipse(&rectview);
}

if (sum_err[1])
{
    dc.SelectObject(&brush[1]);
    dc.Pie(rectview.TopLeft().x, rectview.TopLeft().y, init_x, init_y, x[0], y[0]);
}

if (sum_err[2])
{
    dc.SelectObject(&brush[2]);
    dc.Pie(rectview.TopLeft().x, rectview.TopLeft().y, init_x, init_y, x[1], y[1]);
}

if (sum_err[3])
{
    dc.SelectObject(&brush[3]);
    dc.Pie(rectview.TopLeft().x, rectview.TopLeft().y, init_x, init_y, x[2], y[2]);
}

if (sum_err[4])
{
    dc.SelectObject(&brush[4]);
    dc.Pie(rectview.TopLeft().x, rectview.TopLeft().y, init_x, init_y, x[3], y[3]);
}
    
```

