**Java Term Project**

2301110326 나예은

1. 프로젝트 사전 구성
2. 개발 중 오류 수정 및 해결방안
3. 각 기능 설명 (\*실행 사진 첨부)
4. 데이터베이스
5. 소스 코드
6. **프로젝트 사전 구성**
   1. *개발 전 요구사항*

* 차량번호와 차종류로 입고/출고 할 수 있다.
* 주차자리는 총 20대이며, 승용차와 SUV로 구분된다.
* 승용차: 15대 (+ SUV 주차 자리에 주차 가능)
* SUV: 5대 (+ 승용차 주차 자리에 주차 불가)
* 현재 주차상황을 확인할 수 있다.
* GUI 기능을 지원한다.
* 데이터베이스를 통해 값을 저장하고 사용한다.  
  1. *기능별 고려사항*
* 입고  
  1) 숫자 4자리가 다 입력이 되었는가   
  2) 입력된 번호와 종류가 이미 존재하는가  
  3) 비어 있는 주차 공간을 발견했는가  
  4) SUV주차 공간이 없으면서, SUV가 들어왔는가  
  5) 주차공간이 남아있는가
* 출고  
  1) 숫자 4자리가 다 입력이 되었는가  
  2) 입력된 차량번호가 이미 존재하는가
* 주차장 확인  
  1) 데이터를 어떠한 방식으로 출력할 것인가  
  2) 사용자에게 필요한 데이터가 무엇인가
* 데이터베이스   
  1) 테이블에 어떠한 값을 넣을 것인가  
  2) 어떠한 값을 Primary key로 설정할 것인가  
  3) 데이터베이스에서 값을 어떻게 가져올 것인가  
  4) 데이터베이스에 해당 값이 있는지 어떻게 알 것인가
* Others  
  - GUI 구성을 어떻게 할 것인가

1. **개발 중 오류 수정 및 해결방안**
   1. *GUI 구성*

* 문제: 아래 사진처럼 GUI를 구성한대로 시스템을 개발하던 중 기능 부분에서 주차장 데이터를 다른 GUI를 통해서 가져오기에는 복잡성과 동시에 창 열림으로 인해 오류가 생길 수 있다고 판단

텍스트, 스크린샷, 폰트, 화이트이(가) 표시된 사진

자동 생성된 설명

<초기 GUI 구성안>

* 해결방안: 해당 문제를 해결하기 위해 사용자가 보기 쉽게 한 페이지에 확인할 수 있는 GUI로 재구성

|  |  |
| --- | --- |
| 텍스트, 폰트, 스크린샷, 라인이(가) 표시된 사진  자동 생성된 설명 | 텍스트, 스크린샷, 번호, 디스플레이이(가) 표시된 사진  자동 생성된 설명 |

<재구성한 GUI>

* 주차장 확인을 누른 것과 입고, 출고를 하는 것이 차이가 없어서 주차장 확인버튼을 삭제하는 대신, 프로그램 실행과 동시에 띄우게 만들었다.

텍스트, 스크린샷, 번호, 평행이(가) 표시된 사진

자동 생성된 설명

<최종 GUI>

* 1. 주차장 확인 오류
* 문제: 현재 주차장 상태 확인을 위해 테이블이 표시 되어있는 상태에서 값 로드가 초과된 문제 발생

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

<문제 발생 오류>

* 해결방안: SQL쿼리를 실행한 ‘Statement’나 ‘ResultSet’의 커서 사용한 후에 close() 메서드를 호출하여 명시적으로 닫아주는 작업을 수행함으로써 오류 수정

텍스트, 폰트, 그래픽, 타이포그래피이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 폰트, 라인이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명텍스트, 폰트, 스크린샷, 친필이(가) 표시된 사진

자동 생성된 설명

1. **각 기능 설명 (\*실행 사진 첨부)**

(\*기능 설명 시작 전, GUI배치를 아래 사진의 위치 번호를 참고하여 설명)

라인, 도표, 디자인이(가) 표시된 사진

자동 생성된 설명

<참고: GUI 배치 번호표>

* 처음 실행 화면

|  |  |
| --- | --- |
|  | **GUI** |
| ①: 값이 입력되지 않은 모습 출력  ②: 차 종류 선택 radio 버튼 및 현재 주차된 차종 수, 기능 버튼 출력  - 입고, 출고 버튼을 통해 해당 기능을 수행  ③: 숫자 입력 num pad 출력  ④: 주차장 확인 칸 |
|  | **CLI** |
| 데이터베이스 연결 상태 및 현재의 승용차와 SUV가 데이터베이스에 얼마나 저장 되어있는지 체크하기 위한 값 출력한다. |

* 주차장 확인

|  |  |
| --- | --- |
|  | 프로그램 시작과 동시에 DB와 연결되며, ④칸에 해당 데이터베이스에 저장 되어있는 값들을 출력하여 확인 가능하게 한다.  *? 해당 데이터가 저장되어 있을 경우*  ‘차량번호 (차 종류)’  *? 해당 데이터가 저장되어 있지 않을 경우*  ‘-‘  경우에 따라 외 따옴표에 있는 값을 GUI에 보여준다. |

* ②에 Num Pad를 클릭하여 숫자 입력, 삭제

|  |  |
| --- | --- |
|  | **③ 에서 ‘1234’ 를 클릭 시,**  ① 에서 입력한 값이 표시된다. |
|  | **③ 에서 ‘DEL’ 클릭 시,**  ① 에서 입력된 글자의 맨 마지막 한 글자가 삭제된다.  + 입력된 글자가 없을 시, 삭제 기능 무력화된다. |
|  | **③ 에서 ‘A\_DEL’ 클릭 시,**  ① 에서 입력된 글자 전체가 삭제된다. |

* 입고

|  |  |
| --- | --- |
| Before:    After:    CLI: | **③ 에서 입고 버튼 클릭 시,**  *? ① 에 4자리 값이 있을 경우 (+ After)*  - 입고 확정 메시지를 보여준다.  예 – DB 에 값 저장 / 아니오 – 취소  - ④ 에 입고한 값이 추가된다.  - ① 에 입력된 4자리 값이 초기화된다.  - ③ 에 Now Parked 의 승용차 값이 수정된다.  - 이 후에,  *? 저장할 공간이 있는가*  *? 차 번호가 중복되지 않았는가*  *? 한정된 SUV 주차공간에 걸리지 않는가*  에 대한 심사를 통해 DB에 값을 저장한다.  *? ① 에 4자리 미만의 값이 있을 경우*  - 아래 사진과 같이, “Error: The input is not valid.” 오류 메시지를 화면에 띄운다.    - ① 에 입력된 4자리 값이 초기화된다.  - DB 저장 X  **\* CLI:** 어떠한 SQL문이 들어갔는지, 데이터베이스에서 데이터를 저장할 때 출력한다. 해당 구문이 출력이 된다면 잘 저장된 것이다. |
|  | *? 같은 차량번호를 입력할 경우*  - 왼쪽 사진과 같이, “Error: Already registered vehicle number.” 오류 메시지를 화면에 띄운다.  - ① 에 입력된 4자리 값이 초기화된다.  - DB 저장 X |
|  | *? SUV 차량 입고할 경우*  - ④ 에 SUV 차량이 16~20 공간에 주차된다.  - 이 공간은 승용차와 공유한다.  - 주차 공간이 없을 경우, SUV 차량이 들어갈 수 없다. |
|  | *? SUV 주차 공간이 꽉 찼을 경우*  - 왼쪽 사진과 같이, “Error: No more parking space. (SUV)” 오류 메시지를 화면에 띄운다.  - ① 에 입력된 4자리 값이 초기화된다.  - DB 저장 X |
|  | *? 주차장이 꽉 찼을 경우*  - 아래 사진과 같이, “Error: No more parking space.” 오류 메시지를 화면에 띄운다.    - ① 에 입력된 4자리 값이 초기화된다.  - DB 저장 X |

* 출고

|  |  |
| --- | --- |
| Before:    After: | **③ 에서 입고 버튼 클릭 시,**  *? ① 에 4자리 값이 있을 경우 (+ After)*  - 아래 사진과 같이, 출고 확정 메시지를 보여준다.  예 – DB 에 값 저장 / 아니오 – 취소    - radio 버튼에 선택한 것과 관계없이 차량번호만을 입력해 출고한다.  - ④ 에 출고한 차량번호 값이 사라진다.  - ① 에 입력된 4자리 값이 초기화된다.  - ③ 에 Now Parked 의 승용차 값이 수정된다. |
|  | *? ① 에 4자리 미만의 값이 있을 경우*  - 아래 사진과 같이, “Error: The input is not valid.” 오류 메시지를 화면에 띄운다.    - ① 에 입력된 4자리 값이 초기화된다.  - DB 삭제 X |
|  | *? ① 에 입력된 주차 차량번호가 없는 경우*  - 왼쪽 사진과 같이, “Error: The vehicle is not parked.” 오류 메시지를 화면에 띄운다.  - ① 에 입력된 4자리 값이 초기화된다.  - 주차한 차량이 없어도 해당 시스템에 걸러진다.  - DB 삭제 X |

* Now Parked

|  |  |
| --- | --- |
| 텍스트, 폰트, 화이트, 타이포그래피이(가) 표시된 사진  자동 생성된 설명 | 해당 부분은 승용차와 SUV의 주차 공간이 얼마나 주차되어 있는지를 확인할 수 있다.  승용차는 1~15까지의 자리가 설정되어 있고, 승용차의 주차 공간에 주차된 차를 카운트해 저장된다. 이때, 승용차량은 SUV 주차공간에도 저장할 수 있고, SUV차량은 승용차 주차 공간에 주차할 수 없다.  SUV는 16~20 까지의 자리가 설정되어 SUV주차 공간에 주차된 차를 카운트해 저장된다. |

1. **데이터베이스**

* DataBase Table (PCAR)

|  |  |  |  |
| --- | --- | --- | --- |
| **Primary Key** | **Column Name** | **Data Type** | **Nullable** |
|  | pid | Number (20, 0) | No |
| ✔ | carNum | Varchar2 (10 Byte) | No |
|  | kinds | Varchar2 (10 Byte) | No |

: 데이터베이스 테이블 PCAR 사용한다.

프로그램을 실행하는 동안 아래에 작성되는 SQL문을 이용하여 값을 구한다.

**“SELECT count(\*) as carNum FROM pcar WHERE pid” + str**

-> str에는 “<=20”, “>=16” 등을 넣어서 주차자리를 설정해주고, 해당 차종의 주차자리에 얼만큼 주차되었는지 확인할 때 사용된다.

**“SELECT \* FROM pcar WHERE pid=” + i**

-> i에는 일반 숫자값이 들어가서 데이터베이스에 있는 값들을 반환하여 출력해준다.

**“INSERT INTO pcar VALUES(“ + i + “, ‘” + carNumText.getText() + “’, ‘승용차’)”**

**“INSERT INTO pcar VALUES(“ + i + “, ‘” + carNumText.getText() + “’, ‘SUV’)”**

-> 해당 번호의 차량을 입고할 때 데이터베이스에 Insert문을 사용하여 데이터베이스에 해당 데이터를 생성한다.

**“DELETE FROM pcar WHERE carNum=’” + carNumText.getText() + “’”**

-> 해당 번호의 차량을 출고할 때 데이터베이스에 Delete문을 사용하여 데이터베이스에 해당 데이터를 삭제한다.

1. **소스코드**

* GUI.java

: GUI와 함께 각 기능을 수행하는 메소드가 함께 있다.

|  |
| --- |
| package termProjectParkingSys;  import java.awt.EventQueue;  import javax.swing.JFrame;  import javax.swing.JPanel;  import javax.swing.border.EmptyBorder;  import java.awt.GridLayout;  import javax.swing.JButton;  import javax.swing.AbstractButton;  import javax.swing.ButtonGroup;  import javax.swing.JLabel;  import javax.swing.SwingConstants;  import java.awt.FlowLayout;  import javax.swing.JRadioButton;  import javax.swing.JTextField;  import javax.swing.table.TableColumnModel;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  import java.awt.event.MouseAdapter;  import java.awt.event.MouseEvent;  import java.sql.ResultSet;  import javax.swing.JOptionPane;  import javax.swing.SpringLayout;  import javax.swing.JTable;  import javax.swing.\*;  import javax.swing.table.DefaultTableCellRenderer;  import javax.swing.table.DefaultTableModel;  import javax.swing.table.TableColumn;  import java.sql.ResultSet;  import java.sql.Statement;  import java.awt.Color;  import java.awt.Font;  public class GUI extends JFrame {  private DBController DBC = new DBController();  private static final long serialVersionUID = 1L;  private JPanel contentPane;  private DefaultTableModel tableModel; // DefaultTableModel 추가  private JTable parkingTable;  private JLabel Kinds1Num;  private JLabel Kinds2Num;    // Launch the application.    public static void main(String[] args) {  EventQueue.invokeLater(new Runnable() {  public void run() {  try {  GUI frame = new GUI();  frame.setVisible(true);  } catch (Exception e) {  e.printStackTrace();  }  }  });  }  // Create the frame.    public GUI() {  setBackground(new Color(255, 255, 255));  ResultSet rs;    setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  setBounds(100, 100, 763, 546);  setTitle("Parking System");  contentPane = new JPanel();  contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));  setContentPane(contentPane);  contentPane.setLayout(new GridLayout(2, 3, 0, 0));    JPanel showNum = new JPanel();  showNum.setBackground(new Color(255, 255, 255));  contentPane.add(showNum);  showNum.setLayout(new GridLayout(3, 1, 0, 0));    JPanel gongPanel1 = new JPanel();  gongPanel1.setBackground(new Color(255, 255, 255));  showNum.add(gongPanel1);  gongPanel1.setLayout(new GridLayout(0, 1, 0, 0));    JLabel lblNewLabel = new JLabel("Input your car number (4):");  lblNewLabel.setFont(new Font("돋움", Font.PLAIN, 20));  lblNewLabel.setHorizontalAlignment(SwingConstants.CENTER);  lblNewLabel.setVerticalAlignment(SwingConstants.BOTTOM);  gongPanel1.add(lblNewLabel);    JLabel carNumText = new JLabel("");  carNumText.setFont(new Font("돋움", Font.PLAIN, 65));  carNumText.setBackground(new Color(255, 255, 255));  carNumText.setHorizontalAlignment(SwingConstants.CENTER);  showNum.add(carNumText);    JPanel kinds\_panel = new JPanel();  kinds\_panel.setBackground(new Color(255, 255, 255));      contentPane.add(kinds\_panel);  kinds\_panel.setLayout(new GridLayout(4, 1, 0, 0));    JLabel checkLabelKinds = new JLabel("Check Your Car Kinds:");  checkLabelKinds.setFont(new Font("돋움", Font.PLAIN, 20));  checkLabelKinds.setBackground(new Color(255, 255, 255));  checkLabelKinds.setVerticalAlignment(SwingConstants.BOTTOM);  checkLabelKinds.setHorizontalAlignment(SwingConstants.CENTER);  kinds\_panel.add(checkLabelKinds);    JPanel panel = new JPanel();  kinds\_panel.add(panel);    DBC.startConnection();    // radio 버튼 - kinds  ButtonGroup group = new ButtonGroup();  panel.setLayout(new GridLayout(2, 1, 0, 0));    JPanel panel\_1 = new JPanel();  panel\_1.setBackground(new Color(255, 255, 255));  panel.add(panel\_1);    JRadioButton carKinds1 = new JRadioButton("승용차");  carKinds1.setBackground(new Color(255, 255, 255));  carKinds1.setFont(new Font("돋움", Font.PLAIN, 15));  panel\_1.add(carKinds1);  carKinds1.setSelected(true);    group.add(carKinds1);    JRadioButton carKinds2 = new JRadioButton("SUV");  carKinds2.setBackground(new Color(255, 255, 255));  carKinds2.setFont(new Font("돋움", Font.PLAIN, 15));  panel\_1.add(carKinds2);  group.add(carKinds2);    JPanel panel\_2 = new JPanel();  panel\_2.setBackground(new Color(255, 255, 255));  panel.add(panel\_2);  panel\_2.setLayout(new GridLayout(0, 1, 0, 0));    JLabel showChkKinds = new JLabel("Now Parked:");  showChkKinds.setFont(new Font("돋움", Font.PLAIN, 20));  showChkKinds.setBackground(new Color(255, 255, 255));  showChkKinds.setVerticalAlignment(SwingConstants.BOTTOM);  showChkKinds.setHorizontalAlignment(SwingConstants.CENTER);  panel\_2.add(showChkKinds);    JPanel gongPanel2 = new JPanel();  kinds\_panel.add(gongPanel2);  gongPanel2.setLayout(new GridLayout(2, 1, 0, 0));    JPanel panel\_3 = new JPanel();  panel\_3.setBackground(new Color(255, 255, 255));  gongPanel2.add(panel\_3);    JLabel Kinds1Name = new JLabel("승용차: ");  Kinds1Name.setFont(new Font("돋움", Font.PLAIN, 15));  panel\_3.add(Kinds1Name);    Kinds1Num =new JLabel("");  Kinds1Num.setFont(new Font("돋움", Font.PLAIN, 15));  panel\_3.add(Kinds1Num);    JLabel gongbackT = new JLabel(" ");  panel\_3.add(gongbackT);    JLabel Kinds2Name = new JLabel("SUV: ");  Kinds2Name.setFont(new Font("돋움", Font.PLAIN, 15));  panel\_3.add(Kinds2Name);    Kinds2Num = new JLabel("");  Kinds2Num.setFont(new Font("돋움", Font.PLAIN, 15));  panel\_3.add(Kinds2Num);    getKindsNum();    // 입고  JPanel sys1 = new JPanel();  sys1.setBackground(new Color(255, 255, 255));  gongPanel2.add(sys1);    JButton btnCarIn = new JButton("입고");  btnCarIn.setFont(new Font("돋움", Font.BOLD, 14));  btnCarIn.setForeground(new Color(255, 255, 255));  btnCarIn.setBackground(new Color(26, 55, 87));  btnCarIn.addMouseListener(new MouseAdapter() {  @Override  public void mouseClicked(MouseEvent e) {  if (carNumText.getText().length() == 4) {  try {  String carType = carKinds1.isSelected() ? "승용차" : "SUV";  String sql = "SELECT \* FROM pcar WHERE carnum='"+ carNumText.getText() + "'";  String sql2 = "SELECT \* FROM pcar WHERE pid=";    if (DBC.outDB(sql)) { // 이미 있는 자동차 번호인가?  showErrorDialog("Error: Already registered vehicle number.");  }  else if (carType.equals("SUV")) { // SUV  methodSUVIn(sql2, carNumText);  // 15 ~ 20 주차장을 확인하고 그 부분에서만 집어넣기  }  else { // 승용차  methodCarIn(sql2, carNumText);  }    getKindsNum();  } catch (Exception er) {  er.printStackTrace();  }  }  else { // 올바르지 않는 값일 경우  showErrorDialog("Error: The input is not valid.");  }    carNumText.setText("");  }  });  sys1.add(btnCarIn);    JPanel gongPanel2\_1 = new JPanel();  gongPanel2\_1.setBackground(new Color(255, 255, 255));  kinds\_panel.add(gongPanel2\_1);  gongPanel2\_1.setLayout(new GridLayout(2, 1, 0, 0));      // 주차장 확인  JPanel sys2 = new JPanel();  sys2.setBackground(new Color(255, 255, 255));  gongPanel2\_1.add(sys2);      // 출고  JButton btnCarOut = new JButton("출고");  sys2.add(btnCarOut);  btnCarOut.setFont(new Font("돋움", Font.BOLD, 14));  btnCarOut.setForeground(new Color(255, 255, 255));  btnCarOut.setBackground(new Color(26, 55, 87));    btnCarOut.addMouseListener(new MouseAdapter() {  @Override  public void mouseClicked(MouseEvent e) {  if (carNumText.getText().length() == 4) {    String sql = "SELECT \* FROM pcar WHERE carnum='"+ carNumText.getText() + "'";  if(DBC.outDB(sql)) {  if (showOptionDialog("출고하시겠습니까?", "CarOut")==0) {  sql = "DELETE FROM pcar WHERE carnum='"+ carNumText.getText() + "'";  DBC.ExeQry(sql);  getKindsNum();  loadParkingData();  }  }  else {  showErrorDialog("Error: The vehicle is not parked.");  }  }  else {  showErrorDialog("Error: The input is not valid.");  }    carNumText.setText("");  }  });        // num 버튼 - num\_panel  JPanel num\_panel = new JPanel();  contentPane.add(num\_panel);  num\_panel.setLayout(new GridLayout(4, 3, 0, 0));    JButton btn1 = new JButton("1");  btn1.setFont(new Font("돋움", Font.PLAIN, 22));  btn1.setBackground(new Color(255, 255, 255));  btn1.addMouseListener(new MouseAdapter() {  @Override  public void mouseClicked(MouseEvent e) {  if (carNumText.getText().length() < 4) {  carNumText.setText(carNumText.getText() + "1");  }  }  });  num\_panel.add(btn1);    JButton btn2 = new JButton("2");  btn2.setFont(new Font("돋움", Font.PLAIN, 22));  btn2.setBackground(new Color(255, 255, 255));  btn2.addMouseListener(new MouseAdapter() {  @Override  public void mouseClicked(MouseEvent e) {  if (carNumText.getText().length() < 4) {  carNumText.setText(carNumText.getText() + "2");  }  }  });  num\_panel.add(btn2);    JButton btn3 = new JButton("3");  btn3.setFont(new Font("돋움", Font.PLAIN, 22));  btn3.setBackground(new Color(255, 255, 255));  btn3.addMouseListener(new MouseAdapter() {  @Override  public void mouseClicked(MouseEvent e) {  if (carNumText.getText().length() < 4) {  carNumText.setText(carNumText.getText() + "3");  }  }  });  num\_panel.add(btn3);    JButton btn4 = new JButton("4");  btn4.setFont(new Font("돋움", Font.PLAIN, 22));  btn4.setBackground(new Color(255, 255, 255));  btn4.addMouseListener(new MouseAdapter() {  @Override  public void mouseClicked(MouseEvent e) {  if (carNumText.getText().length() < 4) {  carNumText.setText(carNumText.getText() + "4");  }  }  });  num\_panel.add(btn4);    JButton btn5 = new JButton("5");  btn5.setFont(new Font("돋움", Font.PLAIN, 22));  btn5.setBackground(new Color(255, 255, 255));  btn5.addMouseListener(new MouseAdapter() {  @Override  public void mouseClicked(MouseEvent e) {  if (carNumText.getText().length() < 4) {  carNumText.setText(carNumText.getText() + "5");  }  }  });  num\_panel.add(btn5);    JButton btn6 = new JButton("6");  btn6.setFont(new Font("돋움", Font.PLAIN, 22));  btn6.setBackground(new Color(255, 255, 255));  btn6.addMouseListener(new MouseAdapter() {  @Override  public void mouseClicked(MouseEvent e) {  if (carNumText.getText().length() < 4) {  carNumText.setText(carNumText.getText() + "6");  }  }  });  num\_panel.add(btn6);    JButton btn7 = new JButton("7");  btn7.setFont(new Font("돋움", Font.PLAIN, 22));  btn7.setBackground(new Color(255, 255, 255));  btn7.addMouseListener(new MouseAdapter() {  @Override  public void mouseClicked(MouseEvent e) {  if (carNumText.getText().length() < 4) {  carNumText.setText(carNumText.getText() + "7");  }  }  });  num\_panel.add(btn7);    JButton btn8 = new JButton("8");  btn8.setFont(new Font("돋움", Font.PLAIN, 22));  btn8.setBackground(new Color(255, 255, 255));  btn8.addMouseListener(new MouseAdapter() {  @Override  public void mouseClicked(MouseEvent e) {  if (carNumText.getText().length() < 4) {  carNumText.setText(carNumText.getText() + "8");  }  }  });  num\_panel.add(btn8);    JButton btn9 = new JButton("9");  btn9.setFont(new Font("돋움", Font.PLAIN, 22));  btn9.setBackground(new Color(255, 255, 255));  btn9.addMouseListener(new MouseAdapter() {  @Override  public void mouseClicked(MouseEvent e) {  if (carNumText.getText().length() < 4) {  carNumText.setText(carNumText.getText() + "9");  }  }  });  num\_panel.add(btn9);    JButton btnDelete = new JButton("DEL");  btnDelete.setFont(new Font("돋움", Font.PLAIN, 20));  btnDelete.setBackground(new Color(255, 255, 255));  btnDelete.addMouseListener(new MouseAdapter() {  @Override  public void mouseClicked(MouseEvent e) {  if (carNumText.getText().length() == 0) {  }  else {  String s = carNumText.getText().substring(0, carNumText.getText().length() - 1);  carNumText.setText(s);  }  }  });  num\_panel.add(btnDelete);    JButton btn0 = new JButton("0");  btn0.setFont(new Font("돋움", Font.PLAIN, 22));  btn0.setBackground(new Color(255, 255, 255));  btn0.addMouseListener(new MouseAdapter() {  @Override  public void mouseClicked(MouseEvent e) {  if (carNumText.getText().length() < 4) {  carNumText.setText(carNumText.getText() + "0");  }  }  });  num\_panel.add(btn0);    JButton btnAllDelete = new JButton("A\_DEL");  btnAllDelete.setFont(new Font("돋움", Font.PLAIN, 20));  btnAllDelete.setBackground(new Color(255, 255, 255));  btnAllDelete.addMouseListener(new MouseAdapter() {  @Override  public void mouseClicked(MouseEvent e) {  carNumText.setText("");  }  });  num\_panel.add(btnAllDelete);    JPanel chkAction = new JPanel();  contentPane.add(chkAction);  chkAction.setLayout(new GridLayout(1, 1, 0, 0));      // 주차장 현황 상태 출력 테이블  tableModel = new DefaultTableModel();  parkingTable = new JTable(tableModel) {  @Override  public boolean isCellEditable(int row, int column) {  return false;  }  };    JScrollPane scrollPane = new JScrollPane(parkingTable);  chkAction.add(scrollPane);    loadParkingData();  }  private static int showOptionDialog(String message, String sys) {  int ans = JOptionPane.showConfirmDialog(null, message, sys, JOptionPane.YES\_NO\_OPTION, JOptionPane.PLAIN\_MESSAGE);  return ans;  }    private static void showErrorDialog(String message) {  // JOptionPane을 사용하여 에러 다이얼로그 표시  JOptionPane.showMessageDialog(null, message, "Error", JOptionPane.ERROR\_MESSAGE);  }    private void loadParkingData() {  TableColumnModel columnModel = parkingTable.getColumnModel();    tableModel.setRowCount(0);  tableModel.setColumnCount(0);  tableModel.addColumn("Index");  tableModel.addColumn("차량번호 (종류)");    DefaultTableCellRenderer centerRenderer = new DefaultTableCellRenderer();  centerRenderer.setHorizontalAlignment(JLabel.CENTER);    for (int i = 1; i <= 20; i++) {  String sql = "SELECT \* FROM pcar WHERE pid=" + i;    if (!DBC.outDB(sql)) {  tableModel.addRow(new Object[]{i, "-"});  } else {  String parkInfo = DBC.showParking(i); // 주차 번호 및 차량 정보 조회  tableModel.addRow(new Object[]{i, parkInfo}); // 주차장 테이블에 추가  }  }    for (int i = 0; i < tableModel.getColumnCount(); i++) {  parkingTable.getColumnModel().getColumn(i).setCellRenderer(centerRenderer);  }    columnModel.getColumn(0).setPreferredWidth(5);  parkingTable.setRowHeight(30);  }    private void getKindsNum() {  Kinds1Num.setText(""+ DBC.outNum("<= 15") + "/15");  Kinds2Num.setText(""+ DBC.outNum(">= 16") + "/5");  }    private void methodCarIn(String sql, JLabel carNumText) {  for(int i=1; i<21; i++) {  String sql2 = "select count(\*) as num from pcar where pid<=20";    if (DBC.outNumber(sql2) == 20) { // 꽉 찼을 경우  showErrorDialog("Error: No more parking space.");  break;  }  else if (!DBC.outDB(sql+i)) { // 자리가 비어있을 경우  if (showOptionDialog("입고하시겠습니까?", "CarIn")==0) {  sql = "INSERT INTO pcar VALUES("+ i +", '"+ carNumText.getText() +"', '승용차')";    DBC.ExeQry(sql);  loadParkingData();  }  break;  }  }  }    private void methodSUVIn(String sql, JLabel carNumText) {  for(int i=16; i<21; i++) {  String sql2 = "select count(\*) as num from pcar where pid";    if (DBC.outNumber(sql2 + "<= 20") == 20) {  showErrorDialog("Error: No more parking space.");  break;  }  else if (DBC.outNumber(sql2 + ">=16") == 5) { // SUV 자리가 꽉 찼을 경우  showErrorDialog("Error: No more parking space. (SUV)");  break;  }  else if (!DBC.outDB(sql + i)) { // 자리가 비어있을 경우  if (showOptionDialog("입고하시겠습니까?", "CarIn")==0) {  sql = "INSERT INTO pcar VALUES("+ i +", '"+ carNumText.getText() +"', 'SUV')";    DBC.ExeQry(sql);  loadParkingData();  }  break;  }  }  }  } |

* DBController.java

: 프로그램 시작과 동시에 Database가 연결되며, SQL문을 사용

|  |
| --- |
| package termProjectParkingSys;  import java.io.\*;  import java.sql.\*;  import java.sql.CallableStatement;  public class DBController {  Connection con;  String url="jdbc:oracle:thin:@localhost:1521:xe";  String userid="c##TP2"; /\* 12버전 이상은 c##을 붙인다. \*/  String pwd="termproject";    public DBController () {  try { /\* 드라이버를 찾는 과정 \*/  Class.forName("oracle.jdbc.driver.OracleDriver");  System.out.println ("드라이버 로드 성공");  } catch(ClassNotFoundException e) {  e.printStackTrace();  }  }    public boolean startConnection() {  boolean ret = false;  try /\* 데이터베이스를 연결하는 과정 \*/  {  System.out.println ("데이터베이스 연결 준비 ...");  con=DriverManager.getConnection(url, userid, pwd);  System.out.println ("데이터베이스 연결 성공");  ret = true;  } catch(SQLException e) {  e.printStackTrace();  ret = false;  }    return ret;  }    public void ExeQry(String sql) { // insert, delete 쓸 때  try {  System.out.println(sql);  Statement stmt = con.createStatement();  stmt.execute(sql);  stmt.close();  } catch (Exception e) {  e.printStackTrace();  }  }    public boolean outDB(String sql) {  try {  Statement stmt = con.createStatement();  ResultSet rs = stmt.executeQuery(sql);    if(rs.next()) { // 값이 있을 경우, true 반환  rs.close();  stmt.close();  return true;  } else {  rs.close();  stmt.close();  }  } catch (Exception e) {  e.printStackTrace();  }  return false; // null이면, false 반환  }    public int outNum(String str) {  try {  Statement stmt = con.createStatement();  ResultSet rs = stmt.executeQuery("SELECT count(\*) as carnum FROM pcar WHERE pid"+ str);    if (rs.next()) {  int num = rs.getInt("carnum");    rs.close();  stmt.close();  return num;  } else {  System.out.println("No rows in the result set.");  rs.close();  stmt.close();  }  } catch (Exception e) {  e.printStackTrace();  }  return 0;  }    public int outNumber(String sql) {  try {  Statement stmt = con.createStatement();  ResultSet rs = stmt.executeQuery(sql);    if (rs.next()) {  int num = rs.getInt("num");  rs.close();  stmt.close();  return num;  } else {  System.out.println("No rows in the result set.");  rs.close();  stmt.close();  }  } catch (Exception e) {  e.printStackTrace();  }  return 0;  }    public String showParking(int i) {  try {  Statement stmt = con.createStatement();  String sql = "SELECT \* FROM pcar WHERE pid="+ i;  ResultSet rs = stmt.executeQuery(sql);  String parksql = "";    if (rs.next())  parksql = rs.getString("carnum") + " (" + rs.getString("kinds") + ")";    rs.close();  stmt.close();  return parksql;  } catch(Exception e) {  e.printStackTrace();  }  return "";  }  } |