



라이브러리를 활용한 애플리케이션 구현



한국기술교육대학교
온라인평생교육원

학습내용

- 로또 프로그램
- 회원관리 프로그램

학습목표

- **Swing을 이용하여 로또 프로그램을 작성할 수 있다.**
- **Swing과 JDBC를 이용하여 회원관리 프로그램을 작성할 수 있다.**

로또 프로그램

1 로또 프로그램 개요

① 로또 프로그램 실행 조건

- 1 1 게임당 6개의 정수 값을 난수로 발생
- 2 6개의 정수 값은 1에서 45사이의 값
- 3 6개의 정수 값은 중복 불가
- 4 게임 수는 1~5게임 선택 가능
- 5 Swing을 이용한 GUI 프로그램으로 작성

1 로또 프로그램 개요

② 클래스 구성

클래스	설 명
Lotto	6개의 난수 값을 가지는 1게임을 표현한 클래스
LottoMachine	게임 수를 선택하여 결과를 화면에 표시하는 클래스

로또 프로그램

1 로또 프로그램 개요

③ 클래스 설계

1 Lotto 클래스

구분	메소드 이름	설명
생성자 메소드	Lotto()	6개의 난수 값을 생성
변수	Vector<Integer> lotto	6개의 난수 값을 Integer로 저장하는 Vector 클래스
메소드	Vector<Integer> getNumber()	6개의 난수 값을 Vector 객체로 반환

1 로또 프로그램 개요

③ 클래스 설계

2 LottoMachine 클래스

구분	메소드 이름	설명
생성자 메소드	LottoMachine()	Swing을 이용하여 GUI 구성
변수	JComboBox<Integer> game	게임 수를 선택할 수 있는 콤보 박스
	JButton start	로또 번호를 생성하기 위한 버튼
	JPanel output	생성된 로또 번호를 화면에 표시하기 위한 컨테이너
	JLabel info	메시지 또는 일자와 시간을 표시

로또 프로그램

1 로또 프로그램 개요

③ 클래스 설계

2 LottoMachine 클래스

구분	메소드 이름	설명
메소드	void actionPerformed (ActionEvent e)	start 버튼을 처리하는 이벤트 핸들러 메소드 로또 번호를 생성해서 화면에 표시
내부 클래스	class LinePanel extends JPanel	1 게임의 로또 번호를 화면에 표시하기 위한 클래스

2 로또 프로그램 구현

① Lotto 클래스

1 6개의 난수 값 발생 코드

```

Vector<Integer> lotto = new Vector<Integer>();
Random r = new Random();
while(lotto.size()<6) {
    int number = r.nextInt(46);
    if(number !=0 && !lotto.contains(number)) {
        lotto.add(number);
    }
}
Collections.sort(lotto);

```

로또 프로그램

2 로또 프로그램 구현

① Lotto 클래스

1 6개의 난수 값 발생 코드

난수 6개를 저장할 Vector<integer> 정의

```

Vector<Integer> lotto = new Vector<Integer>();
Random r = new Random();
while(lotto.size()<6) {
    int number = r.nextInt(46);
    if(number !=0 && !lotto.contains(number)) {
        lotto.add(number);
    }
}
Collections.sort(lotto);

```

2 로또 프로그램 구현

① Lotto 클래스

1 6개의 난수 값 발생 코드

```

Vector<Integer> lotto = new Vector<Integer>();
Random r = new Random();
while(lotto.size()<6) {
    int number = r.nextInt(46);
    if(number !=0 && !lotto.contains(number)) {
        lotto.add(number);
    }
}
Collections.sort(lotto);

```

← 난수 발생을 위한 Random 객체 생성

로또 프로그램

2 로또 프로그램 구현

① Lotto 클래스

1 6개의 난수 값 발생 코드

난수 6개가
발생될때까지 반복

```

Vector<Integer> lotto = new Vector<Integer>();
Random r = new Random();
while(lotto.size()<6) {
    int number = r.nextInt(46);
    if(number !=0 && !lotto.contains(number)) {
        lotto.add(number);
    }
}
Collections.sort(lotto);

```

2 로또 프로그램 구현

① Lotto 클래스

1 6개의 난수 값 발생 코드

```

Vector<Integer> lotto = new Vector<Integer>();
Random r = new Random();
while(lotto.size()<6) {
    int number = r.nextInt(46);
    if(number !=0 && !lotto.contains(number)) {
        lotto.add(number);
    }
}
Collections.sort(lotto);

```

← 난수 0부터 45까지 발생

로또 프로그램

2 로또 프로그램 구현

① Lotto 클래스

1 6개의 난수 값 발생 코드

```

Vector<Integer> lotto = new Vector<Integer>();
Random r = new Random();
while(lotto.size()<6) {
    int number = r.nextInt(46);
    if(number != 0 && !lotto.contains(number)) {
        lotto.add(number);
    }
}
Collections.sort(lotto);

```

0은 해당 안됨

2 로또 프로그램 구현

① Lotto 클래스

1 6개의 난수 값 발생 코드

```

Vector<Integer> lotto = new Vector<Integer>();
Random r = new Random();
while(lotto.size()<6) {
    int number = r.nextInt(46);
    if(number != 0 && !lotto.contains(number)) {
        lotto.add(number);
    }
}
Collections.sort(lotto);

```

난수 중복 체크

로또 프로그램

2 로또 프로그램 구현

① Lotto 클래스

1 6개의 난수 값 발생 코드

```
Vector<Integer> lotto = new Vector<Integer>();
Random r = new Random();
while(lotto.size()<6) {
    int number = r.nextInt(46);
    if(number !=0 && !lotto.contains(number)) {
        lotto.add(number); ← 난수 값을 Vector에 저장
    }
}
Collections.sort(lotto);
```

2 로또 프로그램 구현

① Lotto 클래스

1 6개의 난수 값 발생 코드

```
Vector<Integer> lotto = new Vector<Integer>();
Random r = new Random();
while(lotto.size()<6) {
    int number = r.nextInt(46);
    if(number !=0 && !lotto.contains(number)) {
        lotto.add(number);
    }
}
Collections.sort(lotto); ← 컬렉션에 저장된 6개의 값을 정렬
```

로또 프로그램

2 로또 프로그램 구현

② LottoMachine 클래스

1 게임 수 만큼 Lotto 객체 생성

```
Integer count = (Integer)game.getSelectedItem();
ArrayList<Lotto> list = new ArrayList<Lotto>();
for(int i=0; i<count; i++) {
    list.add(new Lotto());
}
```

2 로또 프로그램 구현

② LottoMachine 클래스

1 게임 수 만큼 Lotto 객체 생성

콤보 박스에서 선택한 게임 수를 가져옴

```
Integer count = (Integer)game.getSelectedItem();
ArrayList<Lotto> list = new ArrayList<Lotto>();
for(int i=0; i<count; i++) {
    list.add(new Lotto());
}
```

로또 프로그램

2 로또 프로그램 구현

② LottoMachine 클래스

1 게임 수 만큼 Lotto 객체 생성

```
Integer count = (Integer)game.getSelectedItem();
ArrayList<Lotto> list = new ArrayList<Lotto>();
for(int i=0; i<count; i++) {
    list.add(new Lotto());
}
```

Lotto 객체를 저장할 ArrayList<integer> 정의

2 로또 프로그램 구현

② LottoMachine 클래스

1 게임 수 만큼 Lotto 객체 생성

```
Integer count = (Integer)game.getSelectedItem();
ArrayList<Lotto> list = new ArrayList<Lotto>();
for(int i=0; i<count; i++) { ← 게임 수 만큼 반복
    list.add(new Lotto());
}
```

로또 프로그램

2 로또 프로그램 구현

② LottoMachine 클래스

① 게임 수 만큼 Lotto 객체 생성

```
Integer count = (Integer)game.getSelectedItem();
ArrayList<Lotto> list = new ArrayList<Lotto>();
for(int i=0; i<count; i++) {
    list.add(new Lotto());
}
```

← Lotto 객체를 생성하여 ArrayList에 추가

2 로또 프로그램 구현

② LottoMachine 클래스

② 로또 번호를 화면에 출력

```
for(Lotto lotto : list) {
    LinePanel line = new LinePanel(lotto.getNumber());
    output.add(line);
}
```

로또 프로그램

2 로또 프로그램 구현

② LottoMachine 클래스

2 로또 번호를 화면에 출력

ArrayList 객체
↓

```
for(Lotto lotto : list) {
    LinePanel line = new LinePanel(lotto.getNumber());
    output.add(line);
}
```

2 로또 프로그램 구현

② LottoMachine 클래스

2 로또 번호를 화면에 출력

Lotto 객체
↓

```
for(Lotto lotto : list) {
    LinePanel line = new LinePanel(lotto.getNumber());
    output.add(line);
}
```

로또 프로그램

2 로또 프로그램 구현

② LottoMachine 클래스

2 로또 번호를 화면에 출력

```
for(Lotto lotto : list) {
    LinePanel line = new LinePanel(lotto.getNumber());
    output.add(line);
}
```

로또 번호를 가져옴

2 로또 프로그램 구현

② LottoMachine 클래스

2 로또 번호를 화면에 출력

```
LinePanel 객체 생성 → for(Lotto lotto : list) {
    LinePanel line = new LinePanel(lotto.getNumber());
    output.add(line);
}
```

로또 프로그램

2 로또 프로그램 구현

② LottoMachine 클래스

2 로또 번호를 화면에 출력

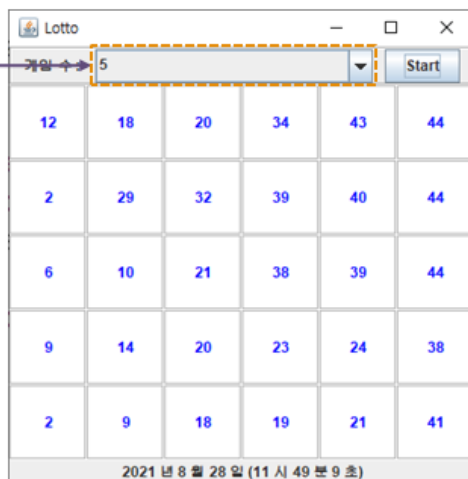
```
for(Lotto lotto : list) {
    LinePanel line = new LinePanel(lotto.getNumber());
    output.add(line);
}
```

output패널에 LinePanel 객체 추가

2 로또 프로그램 구현

③ 프로그램 실행 화면

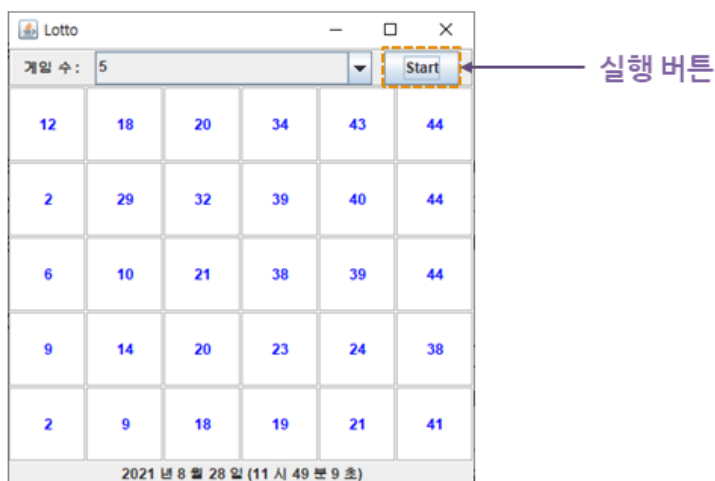
게임 수를 선택하는 콤보 박스



로또 프로그램

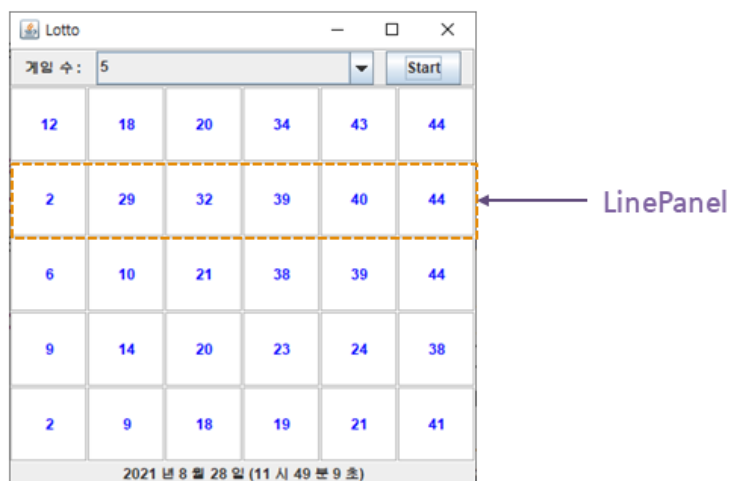
2 로또 프로그램 구현

③ 프로그램 실행 화면



2 로또 프로그램 구현

③ 프로그램 실행 화면



로또 프로그램

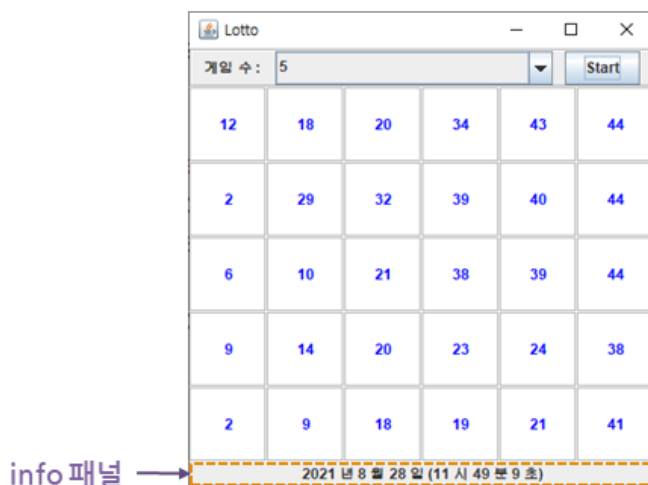
2 로또 프로그램 구현

③ 프로그램 실행 화면



2 로또 프로그램 구현

③ 프로그램 실행 화면



로또 프로그램



라이브러리를 활용한 애플리케이션 구현

로또 프로그램



실습하기



라이브러리를 활용한 애플리케이션 구현

로또 프로그램

실습순서

1. 로또 프로그램 실습



유의사항

- JDK와 이클립스를 설치 완료된 후에 실습이 가능
- 본인이 원하는 작업 폴더를 미리 정해 놓은 다음 실습하기
- 작업 폴더는 C드라이브에 지정하기 보다는 D드라이브나 외장하드디스크를 활용하기를 추천



※ 제공되는 실습 코드를 다운받아 실습해보시기 바랍니다.

회원관리 프로그램

1 회원관리 프로그램 개요

① 회원관리 프로그램 실행 조건

- 1 회원정보(성명, 나이, 전화번호) 관리
- 2 회원정보의 추가, 수정, 삭제, 검색 가능
- 3 회원정보를 데이터베이스에 저장 및 불러오기

1 회원관리 프로그램 개요

② 클래스 구성

클래스	설 명
DBMember	GUI 화면 구성, 데이터베이스 연결
DBInsert	회원 신규 등록
DBManager	회원 정보 수정 및 삭제
DBList	회원 목록

회원관리 프로그램

1 회원관리 프로그램 개요

③ 클래스 설계

1 DBMember 클래스

JFrame을 상속

구분	메소드 이름	설 명
생성자 메소드	DBMember()	데이터베이스 연결 및 GUI 화면 구성
변수	Connection db	데이터베이스 연결 객체
메소드	void stateChanged(ChangeEvent e)	JTabbedPane 이벤트 핸들러 메소드 “회원 목록” 탭의 이벤트 처리

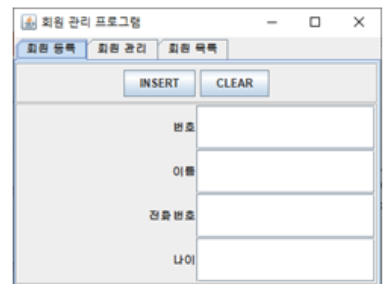
1 회원관리 프로그램 개요

③ 클래스 설계

2 DBInsert 클래스

신규 회원 등록 클래스

JPanel을 상속



회원관리 프로그램

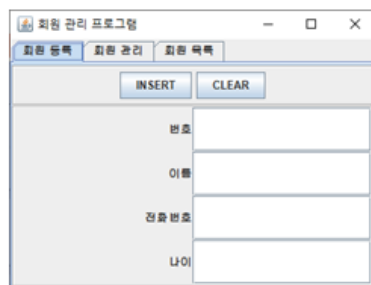
1 회원관리 프로그램 개요

③ 클래스 설계

2 DBInsert 클래스

신규 회원 등록 클래스

JPanel을 상속



구분	메소드 이름	설명
생성자 메소드	DBInsert()	회원 등록 GUI 화면 구성
메소드	void actionPerformed(ActionEvent e)	데이터베이스에 회원 등록

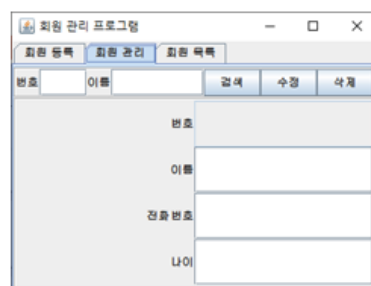
1 회원관리 프로그램 개요

③ 클래스 설계

3 DBManage 클래스

회원 관리(수정 및 삭제) 클래스

JPanel을 상속



구분	메소드 이름	설명
생성자 메소드	DBManage()	회원 관리 GUI 화면 구성
메소드	void actionPerformed(ActionEvent e)	데이터베이스에 회원 정보 수정 및 삭제

회원관리 프로그램

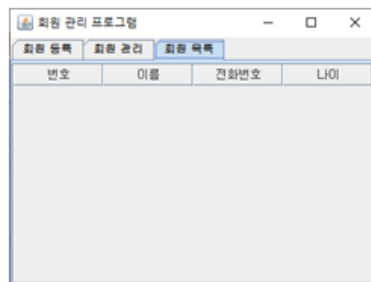
1 회원관리 프로그램 개요

③ 클래스 설계

4 DBList 클래스

회원 목록 클래스

JPanel을 상속



구분	메소드 이름	설명
생성자 메소드	DBList()	회원 목록 GUI 화면 구성 : JTable
내부 클래스	MyTableModel	회원 정보를 JTable에 연결하는 모델 클래스

2 회원관리 프로그램 구현

① MariaDB 설정

1 데이터베이스 생성

MySQL Client 실행

이름 : member

명령 : create database member;

회원관리 프로그램

2 회원관리 프로그램 구현

① MariaDB 설정

2 테이블 생성

데이터베이스 연결 후 테이블 생성

이름 : info

명령 :

```
connect member;
create table info (number int not null primary key,
hname varchar(20) not null, telno varchar(15) not null,
age int not null);
```

2 회원관리 프로그램 구현

② 실행화면

회원 등록(DBInsert)

회원 정보 입력

회원관리 프로그램

2 회원관리 프로그램 구현

② 실행화면

회원 등록(DBInsert)

회원 등록

2 회원관리 프로그램 구현

② 실행화면

회원 관리(DBManage)

회원 관리

회원관리 프로그램

2 회원관리 프로그램 구현

② 실행화면

회원 관리(DBManage)

검색하고자 하는
번호 또는 이름 입력

2 회원관리 프로그램 구현

② 실행화면

회원 관리(DBManage)

회원관리 프로그램

2 회원관리 프로그램 구현

② 실행화면

회원 관리(DBManage)

번호는 수정 불가

번호 1

이름 김장군

전화 번호 010-111-1111

나이 32

2 회원관리 프로그램 구현

② 실행화면

회원 관리(DBManage)

수정 내용 입력

번호 1

이름 김장군

전화 번호 010-111-1111

나이 32

회원관리 프로그램

2 회원관리 프로그램 구현

② 실행화면

회원 관리(DBManage)

회원 정보
수정 또는 삭제

2 회원관리 프로그램 구현

② 실행화면

회원 목록(DBList)

번호	이름	전화번호	나이
1	김장군	010-111-1111	32
2	최왕자	010-222-2222	45
3	이왕비	010-333-3333	42
4	박왕자	010-444-4444	20

전체 회원 정보를
데이터베이스에서 가져옴

회원관리 프로그램

2 회원관리 프로그램 구현

② 실행화면

회원 목록(DBList)

JTable →

번호	이름	전화번호	나이
1	김장군	010-111-1111	32
2	최왕자	010-222-2222	45
3	이왕비	010-333-3333	42
4	박왕자	010-444-4444	20

회원관리 프로그램



라이브러리를 활용한 애플리케이션 구현

회원 관리 프로그램



라이브러리를 활용한 애플리케이션 구현

회원 관리 프로그램

실습순서

1. 회원 관리 프로그램 실습



유의사항

- JDK와 이클립스를 설치 완료된 후에 실습이 가능
- 본인이 원하는 작업 폴더를 미리 정해 놓은 다음 실습하기
- 작업 폴더는 C드라이브에 지정하기 보다는 D드라이브나 외장하드디스크를 활용하기를 추천



※ 제공되는 실습 코드를 다운받아 실습해보시기 바랍니다.

응용문제

라이브러리를 활용한 애플리케이션 구현 응용문제

다음 실행화면과 조건에 맞게 프로그램을 작성하시오.

조건

- 1 “회원 목록” 실행화면에서 테이블을 직접 수정하면, 데이터베이스가 수정되도록 프로그램을 수정하시오.
- 2 전체 목록 보기 버튼 추가

프로젝트명: DBMember

제공되는 실습 소스코드를 다운받아 실습해보시기 바랍니다.

라이브러리를 활용한 애플리케이션 구현 응용문제

실행화면

회원 관리 프로그램			
회원 등록 회원 관리 회원 목록			
번호	이름	전화번호	나이
1	김장군	010-111-1111	32
2	최왕자	010-222-2222	45
3	이왕비	010-333-3333	42
4	박왕자	010-444-4444	20