

TOY PLAYER

JAVA TOY PROJECT



유세영

목차

01

주제 &
프로젝트 설계

02

기능 소개

03

시연

04

소감

01

주제
&
프로젝트 설계

주제

VOL 1

TOY PROJECT?

TOY PLAYER

주제

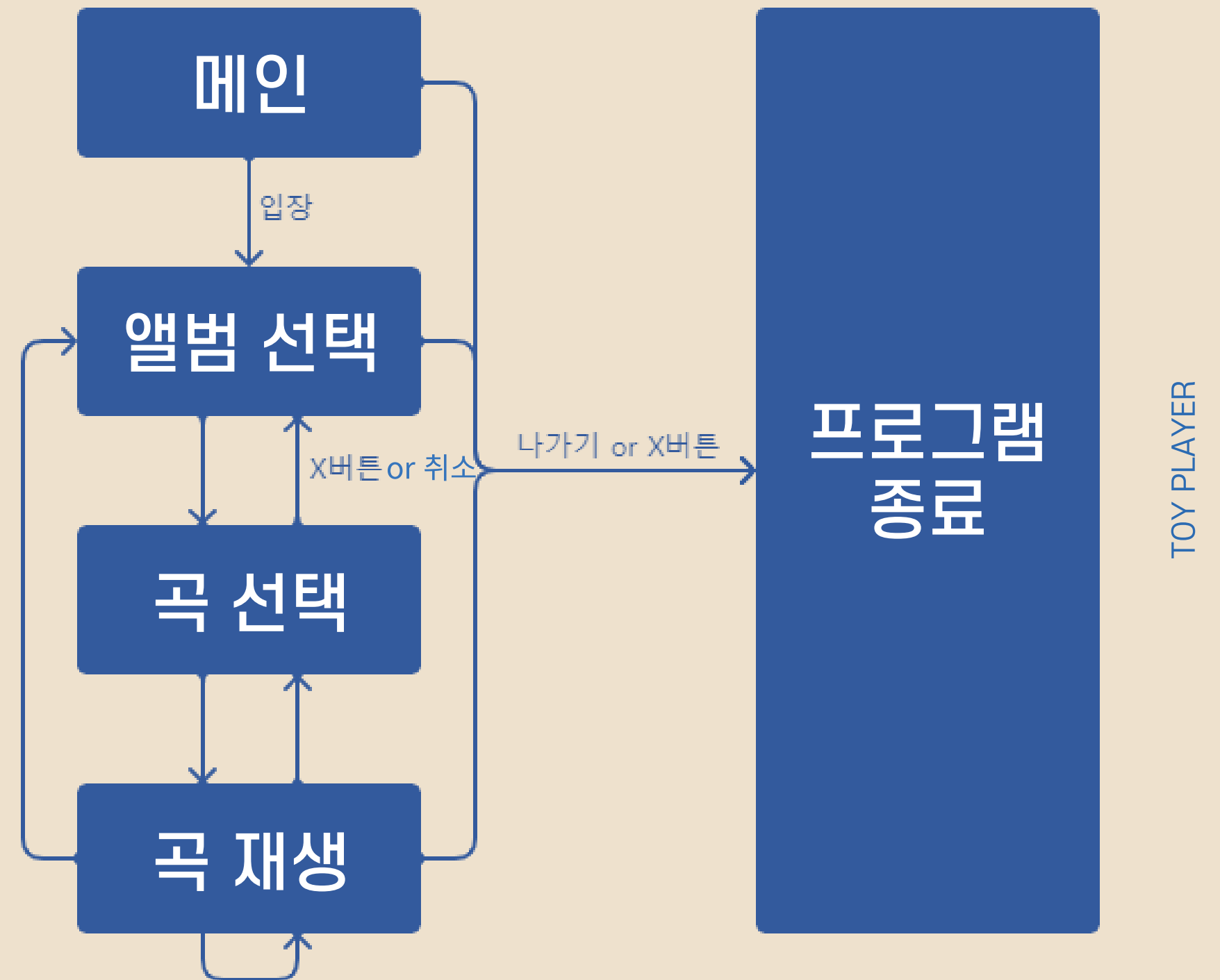
VOL 1

TOY PLAYER !

TOY PLAYER

프로젝트 설계

- 총 네 가지 화면
- 메인, 앨범 선택, 곡 재생
⇒ `JOptionPane.showOptionDialog()` 사용
- 곡 선택
⇒ `JOptionPane.showInputDialog()`
 콤보박스 사용
- 총 다섯 클래스



02

기능 소개

기능 소개

- 첫 번째 클래스
⇒ Jsoup 라이브러리를 활용한 Crawler
- import classes : ArrayList, IOException 등

crawling

```
public class Crawler {
    private static Crawler cr;
    private ArrayList<String> albumUrl = new ArrayList<String>(); // 크롤링 대상 url 리스트
    private String albumListUrl = "https://music.bugs.co.kr/artist/22751/albums?type=RELEASE&sort=R&row=70&page=1&roleCode=0&group=";
    private String album = ".info .albumTitle";
    private String track = ".list .info .title.ellipsis";

    private Crawler() { // 기본 생성자 - 크롤링 대상 url 지정
        albumUrl.add("https://www.genie.co.kr/detail/albumInfo?axnm=15034231");
        albumUrl.add("https://www.genie.co.kr/detail/albumInfo?axnm=15035038");
        albumUrl.add("https://www.genie.co.kr/detail/albumInfo?axnm=15035538");
        albumUrl.add("https://www.genie.co.kr/detail/albumInfo?axnm=15036667");
        albumUrl.add("https://www.genie.co.kr/detail/albumInfo?axnm=15025539");
        albumUrl.add("https://www.genie.co.kr/detail/albumInfo?axnm=54420313");
        albumUrl.add("https://www.genie.co.kr/detail/albumInfo?axnm=80479080");
    }

    public static Crawler getInstance() { // 싱글톤 객체 만들기
        if (cr == null) {
            cr = new Crawler();
        }
        return cr;
    }
}
```

getter, setter

```
public ArrayList<String> getAlbumUrl() { //getter  
    return albumUrl;  
}  
  
public String getAlbumListUrl() {  
    return albumListUrl;  
}
```

Singleton

```
public class Crawler {  
    private static Crawler cr;  
    private ArrayList<String> albumUrl = new ArrayList<String>(); // 크롤링 대상 url 리스트  
    private String albumListUrl = "https://music.bugs.co.kr/artist/22751/albums?type=RELEASE&sort=R&row=70&page=1&roleCode=0&group=";  
    private String album = ".info .albumTitle";  
    private String track = ".list .info .title.ellipsis";  
  
    private Crawler() { // 기본 생성자, 크롤링 대상 url 지정  
        albumUrl.add("https://www.genie.co.kr/detail/albumInfo?axnm=15034231");  
        albumUrl.add("https://www.genie.co.kr/detail/albumInfo?axnm=15035038");  
        albumUrl.add("https://www.genie.co.kr/detail/albumInfo?axnm=15035538");  
        albumUrl.add("https://www.genie.co.kr/detail/albumInfo?axnm=15036667");  
        albumUrl.add("https://www.genie.co.kr/detail/albumInfo?axnm=15025539");  
        albumUrl.add("https://www.genie.co.kr/detail/albumInfo?axnm=54420313");  
        albumUrl.add("https://www.genie.co.kr/detail/albumInfo?axnm=80479080");  
    }  
  
    public static Crawler getInstance() { // 싱글톤 객체 만들기  
        if (cr == null) {  
            cr = new Crawler();  
        }  
        return cr;  
    }  
}
```

try-catch문

```
public Document connect(String url) { // 파싱
    try {
        Document document = Jsoup.connect(url).timeout(5000).maxBodySize(0).get();
        return document;
    } catch (IOException e) {
        e.printStackTrace();
    }
    return null;
}
```

String 클래스 - replace() 메서드

```
public String[] getTrack(Document doc) { // 트랙 이름 데이터 가져오기
    Elements elements = doc.select(track);
    String[] result = new String[elements.size()];
    for (int i = 0; i < result.length; i++) {
        result[i] = elements.get(i).text().replace("?", ""); // 파일명에 "?"가 올 수 없으므로 "?" 제거
    }
    return result;
}
```

기능 소개

- 두 번째 클래스
⇒ SetData
- import classes : ArrayList, ImageIcon 등

Singleton, getter

VOL 1

```
private SetData() {}

public static SetData getInstance() { // 싱글톤 객체 만들기
    if (sd == null) {
        sd = new SetData();
    }
    return sd;
}
```

```
private static SetData sd;
```

```
private ArrayList<Document> doc;
```

```
private String[] main; // 첫 번째 화면 선택지 (입장 여부)
private String[] whichAlbum; // 두 번째 화면 선택지 (앨범 선택)
private String[][] whichTrack; // 세 번째 화면 선택지 (트랙 선택)
private String[] player; // 네 번째 화면 선택지 (트랙 선택)

private ImageIcon mainImg; // 첫 번째 화면 이미지
private ImageIcon albumChoiceImg; // 두 번째 화면 이미지
private ImageIcon playerImg; // 네 번째 화면 이미지
```

```
public ArrayList<Document> getDoc() { //getter
    return doc;
}

public String[] getMain() {
    return main;
}

public String[] getWhichAlbum() {
    return whichAlbum;
}

public String[][] getWhichTrack() {
    return whichTrack;
}

public String[] getPlayer() {
    return player;
}

public ImageIcon getMainImg() {
    return mainImg;
}

public ImageIcon getAlbumChoiceImg() {
    return albumChoiceImg;
}

public ImageIcon getPlayerImg() {
    return playerImg;
}
```

TOY PLAYER

ArrayList

VOL 1

```
public ArrayList<Document> getDoc() { //getter
    return doc;
}

public String[] getMain() {
    return main;
}

public String[] getWhichAlbum() {
    return whichAlbum;
}

public String[][] getWhichTrack() {
    return whichTrack;
}

public String[] getPlayer() {
    return player;
}

public ImageIcon getMainImg() {
    return mainImg;
}

public ImageIcon getAlbumChoiceImg() {
    return albumChoiceImg;
}

public ImageIcon getPlayerImg() {
    return playerImg;
}
```

```
private SetData() {}
```

```
public static SetData getInstance() { // 싱글톤 객체 만들기
    if (sd == null) {
        sd = new SetData();
    }
    return sd;
}
```

```
private static SetData sd;
```

```
private ArrayList<Document> doc;
```

```
private String[] main; // 첫 번째 화면 선택지 (입장 여부)
private String[] whichAlbum; // 두 번째 화면 선택지 (앨범 선택)
private String[][] whichTrack; // 세 번째 화면 선택지 (트랙 선택)
private String[] player; // 네 번째 화면 선택지 (트랙 선택)
```

```
private ImageIcon mainImg; // 첫 번째 화면 이미지
private ImageIcon albumChoiceImg; // 두 번째 화면 이미지
private ImageIcon playerImg; // 네 번째 화면 이미지
```

TOY PLAYER

method()

```
public void setDoc() {
    doc = new ArrayList<Document>(); // html 리스트
    for (String v : cr.getAlbumUrl()) { // html 파싱하기
        doc.add(cr.connect(v));
    }
    doc.add(cr.connect(cr.getAlbumListUrl()));
    System.out.println("파싱한 페이지 개수 : " + doc.size() + "\n");
}

public void setText() {
    main = new String[] { "입장하기", "나가기" };

    whichAlbum = new String[cr.getAlbumUrl().size()];
    whichAlbum = cr.getAlbum(doc.get(doc.size() - 1)); // 앨범 이름 데이터 넣기

    whichTrack = new String[whichAlbum.length][];
    for (int i = 0; i < whichTrack.length; i++) { // 앨범별로 트랙 데이터 넣기, 앨범이름& 트랙 데이터 확인
        whichTrack[i] = cr.getTrack(doc.get(i));
        System.out.println(whichAlbum[i] + " 트랙 : ");
        for (String v : whichTrack[i]) {
            System.out.println(v);
        }
        System.out.println();
    }

    player = new String[] { "다시 재생", "앨범 선택", "곡 선택", "나가기" };
}

public void setImg(Document doc) {
    mainImg = new ImageIcon("src/img/main.gif");
    albumChoiceImg = new ImageIcon("src/img/albumChoice.jpg");
    playerImg = new ImageIcon("src/img/player.gif");
}
}
```

logging

VOL 1

```
public void setDoc() {
    doc = new ArrayList<Document>(); // html 리스트
    for (String v : cr.getAlbumUrl()) { // html 파싱하기
        doc.add(cr.connect(v));
    }
    doc.add(cr.connect(cr.getAlbumListUrl()));
    System.out.println("파싱한 페이지 개수 : " + doc.size() + "\n");

    public void setText() {
        main = new String[] { "입장하기", "나가기" };

        whichAlbum = new String[cr.getAlbumUrl().size()];
        whichAlbum = cr.getAlbum(doc.get(doc.size() - 1)); // 앨범 이름 데이터 넣기

        whichTrack = new String[whichAlbum.length][];
        for (int i = 0; i < whichTrack.length; i++) { // 앨범별로 트랙 데이터 넣기, 앨범이름& 트랙 데이터 확인
            whichTrack[i] = cr.getTrack(doc.get(i));
            System.out.println(whichAlbum[i] + " 트랙 : ");
            for (String v : whichTrack[i]) {
                System.out.println(v);
            }
            System.out.println();
        }

        player = new String[] { "다시 재생", "앨범 선택", "곡 선택", "나가기" };
    }

    public void setImg(Document doc) {
        mainImg = new ImageIcon("src/img/main.gif");
        albumChoiceImg = new ImageIcon("src/img/albumChoice.jpg");
        playerImg = new ImageIcon("src/img/player.gif");
    }
}
```

TOY PLAYER

기능 소개

- 세 번째 클래스
⇒ Screen
- import classes : JOptionPane, ImageIcon

JOptionPane

```
public static int show1(ImageIcon img, String[] arr) { // 메인 화면
    int result = JOptionPane.showOptionDialog(null, "", "  toy player 🎵", JOptionPane.DEFAULT_OPTION,
        JOptionPane.PLAIN_MESSAGE, img, arr, null);
    return result;
}

public static int show2(ImageIcon img, String[] arr) { // 앨범 선택 화면
    int result = JOptionPane.showOptionDialog(null, "", "  앨범 선택 🎧 🔊", JOptionPane.DEFAULT_OPTION,
        JOptionPane.PLAIN_MESSAGE, img, arr, null);
    return result;
}

public static String show3(String album, String[] arr) { // 트랙 선택 화면
    String result = (String) JOptionPane.showInputDialog(null, "", "  " + album + " 🔊", JOptionPane.PLAIN_MESSAGE,
        new ImageIcon("src/img/album_" + album + ".jpg"), arr, null);
    return result;
}

public static int show4(String track, ImageIcon img, String[] arr) { // 재생 화면
    int result = JOptionPane.showOptionDialog(null, "", "  " + track + " 🎵", JOptionPane.DEFAULT_OPTION,
        JOptionPane.PLAIN_MESSAGE, img, arr, null);
    return result;
}
```

static

```
public static int show1(ImageIcon img, String[] arr) { // 메인 화면
    int result = JOptionPane.showOptionDialog(null, "", " toy player 🎵", JOptionPane.DEFAULT_OPTION,
        JOptionPane.PLAIN_MESSAGE, img, arr, null);
    return result;
}

public static int show2(ImageIcon img, String[] arr) { // 앨범 선택 화면
    int result = JOptionPane.showOptionDialog(null, "", " 앨범 선택 🎧 🔊", JOptionPane.DEFAULT_OPTION,
        JOptionPane.PLAIN_MESSAGE, img, arr, null);
    return result;
}

public static String show3(String album, String[] arr) { // 트랙 선택 화면
    String result = (String) JOptionPane.showInputDialog(null, "", " " + album + " 🔊", JOptionPane.PLAIN_MESSAGE,
        new ImageIcon("src/img/album_" + album + ".jpg"), arr, null);
    return result;
}

public static int show4(String track, ImageIcon img, String[] arr) { // 재생 화면
    int result = JOptionPane.showOptionDialog(null, "", " " + track + " 🎵", JOptionPane.DEFAULT_OPTION,
        JOptionPane.PLAIN_MESSAGE, img, arr, null);
    return result;
}
```

기능 소개

- 네 번째 클래스
⇒ Music
- import classes : BufferedInputStream, FileInputStream, Player

Singleton

```
private static Music music;
```

```
private String fileName;
```

```
private Player p;
```

```
private Music(String fileName) { // 노래 제목 설정  
    this.fileName = fileName;  
}
```

```
public static Music getInstance(String fileName) { // 싱글톤 객체 만들기  
    music = new Music(fileName);  
    return music;  
}
```

Thread

```
public void play() {
    try {
        FileInputStream fis = new FileInputStream(fileName);
        BufferedInputStream bis = new BufferedInputStream(fis); // 해당 파일을 버퍼에 담아서 가져올 수 있도록 함
        p = new Player(bis); // 해당 파일을 담을 수 있도록 함
    } catch (Exception e) {
        e.printStackTrace();
    }

    new Thread() {
        public void run() {
            try {
                p.play(); // 곡 재생
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    }.start();
}
```


기능 소개

- 다섯 번째 클래스
⇒ Main
- main() 메서드 포함

data setting

```
SetData sd = SetData.getInstance(); // 데이터 세팅  
sd.setDoc();  
sd.setText();  
sd.setImg(sd.getDoc().get(sd.getDoc().size() - 1));
```

```
int choice1_2Num = 0; // 선택지 저장공간  
String choice3 = null;  
int choice4Num = 0;  
boolean flag = true; // 플래그
```

화면 구성

```
choice1_2Num = Screen.show1(sd.getMainImg(), sd.getMain()); // 메인 화면

if (choice1_2Num == 0) { // 입장하기 눌렀을 때
    while (true) {
        if (flag) {
            choice1_2Num = Screen.show2(sd.getAlbumChoiceImg(), sd.getWhichAlbum()); // 앨범 선택 화면
            if (choice1_2Num == -1) break;
        }

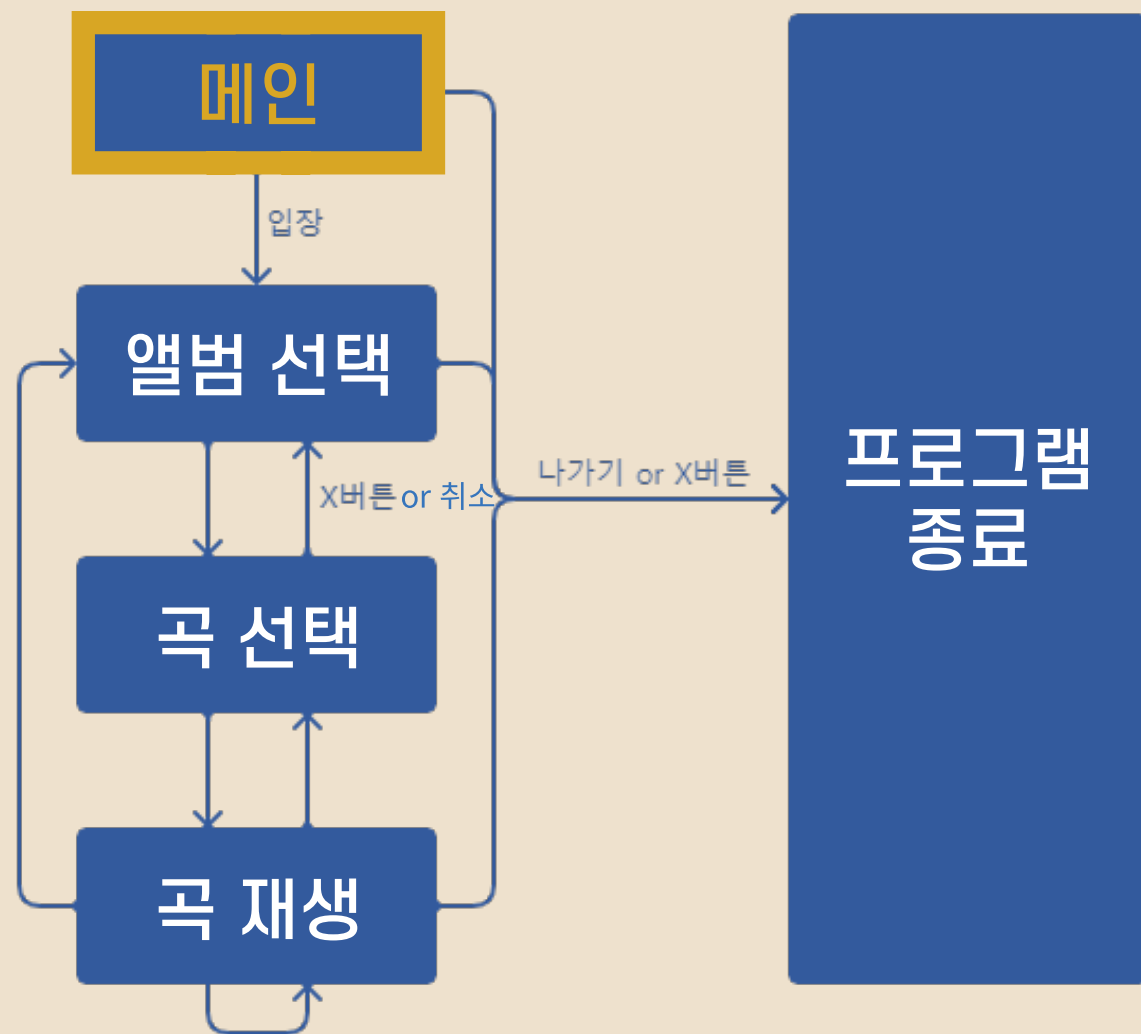
        choice3 = Screen.show3(sd.getWhichAlbum()[choice1_2Num], sd.getWhichTrack()[choice1_2Num]); // 트랙 선택 화면
        if (choice3 == -1) {
            flag = true;
            continue;
        }
        Music music = Music.getInstance("src/music/" + sd.getWhichAlbum()[choice1_2Num] + "/" + choice3 + ".mp3");

        do {
            music.play();
            flag = false;
            choice4Num = Screen.show4(choice3, sd.getPlayerImg(), sd.getPlayer()); // 재생 화면
            if (choice4Num == 0) flag = true;
            music.close();
        } while (flag);

        if (choice4Num == 1) flag = true;
        else if (choice4Num == -1 || choice4Num == 3) break;
    }
}
```

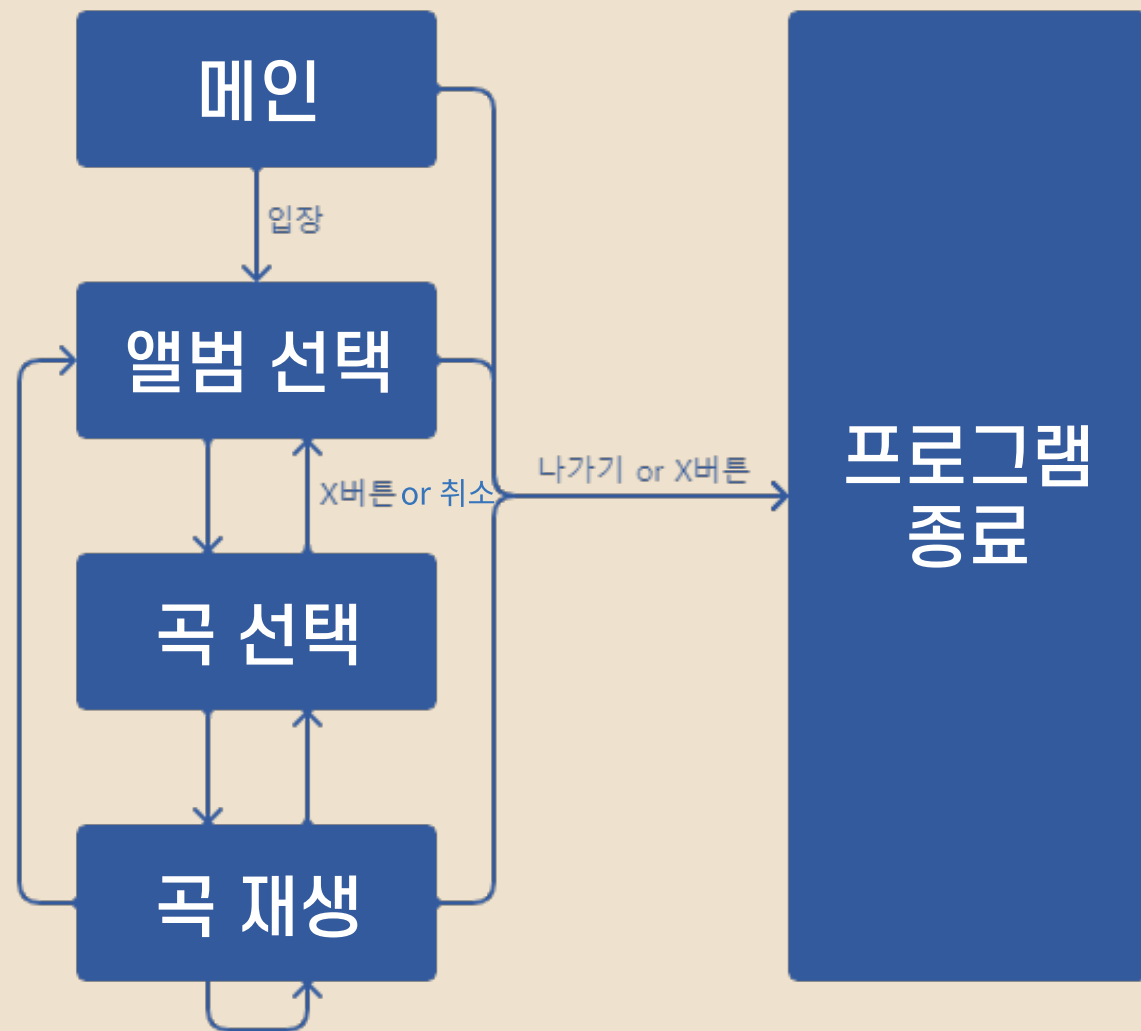
제어문, flag

- `flag = true;`
- 선택지
⇒ 입장하기, 나가기, X창



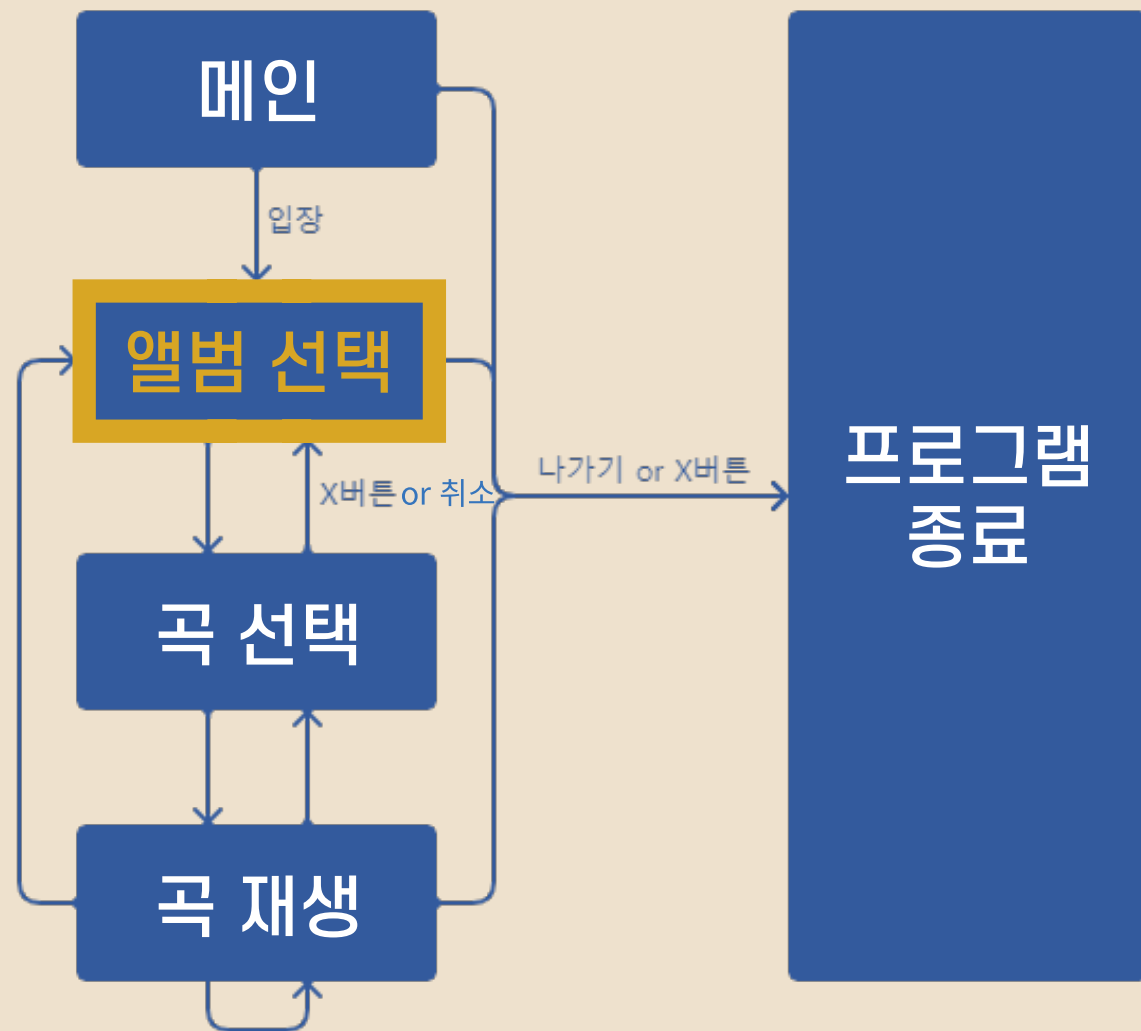
제어문, flag

- `flag = true;`
- `while (true) { }`
⇒ break문 만날때까지 무한 반복



제어문, flag

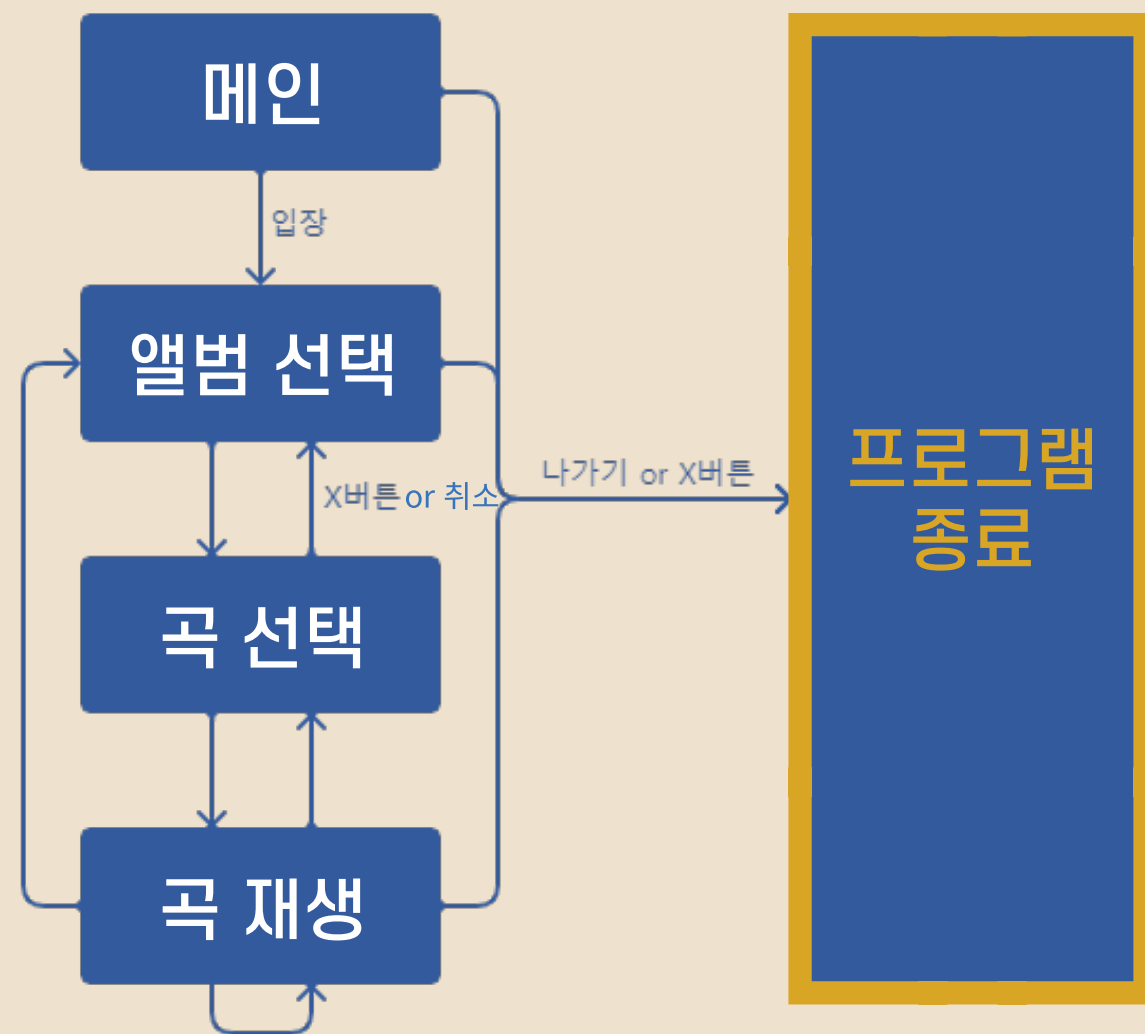
- flag = true;
- 선택지
⇒ 앨범 목록, X창



제어문, flag

- flag = true;
- "X창" 선택 시 프로그램 종료

VOL 1

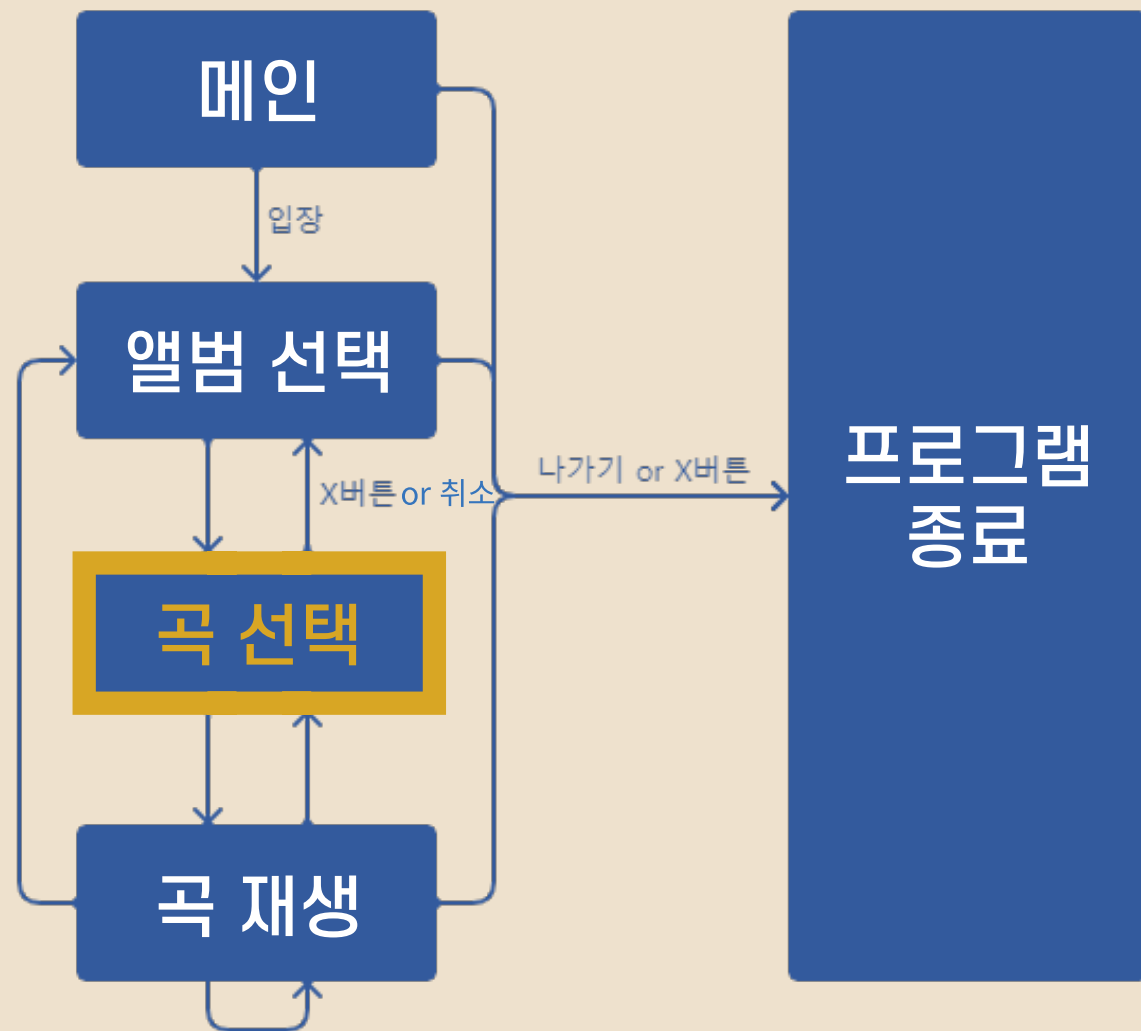


TOY PLAYER



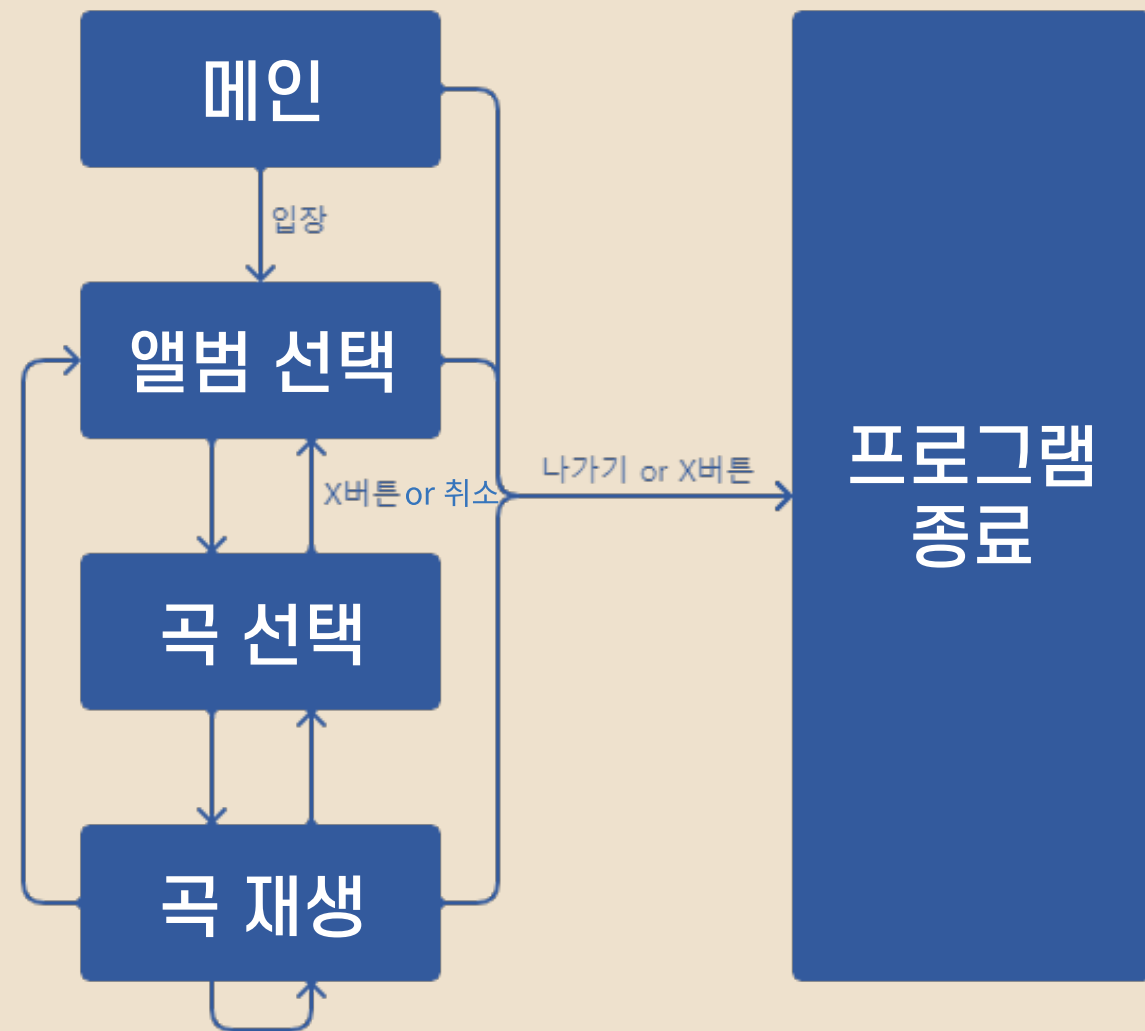
제어문, flag

- flag = true / false;
- 선택지
⇒ 곡 선택 후 확인, 취소, X창



제어문, flag

- flag = true;
(true / false → true)
- "취소" or "X창" 선택 시
continue문 수행



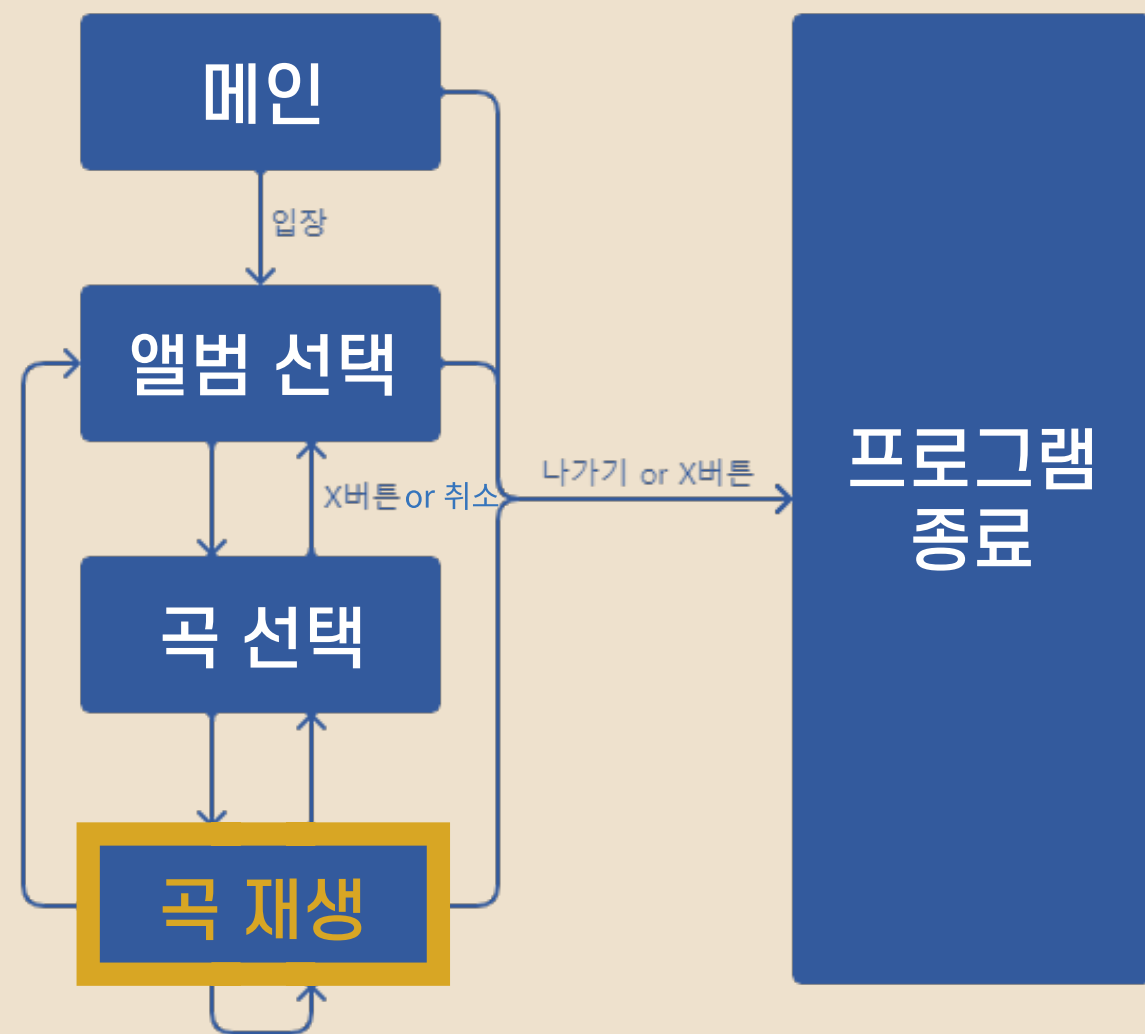
VOL 1

TOY PLAYER



제어문, flag

- `flag = false;`
(`true / false` → `false`)
- 선택지 ⇒ 다시 재생, 앨범 선택, 곡 선택, 나가기, X창



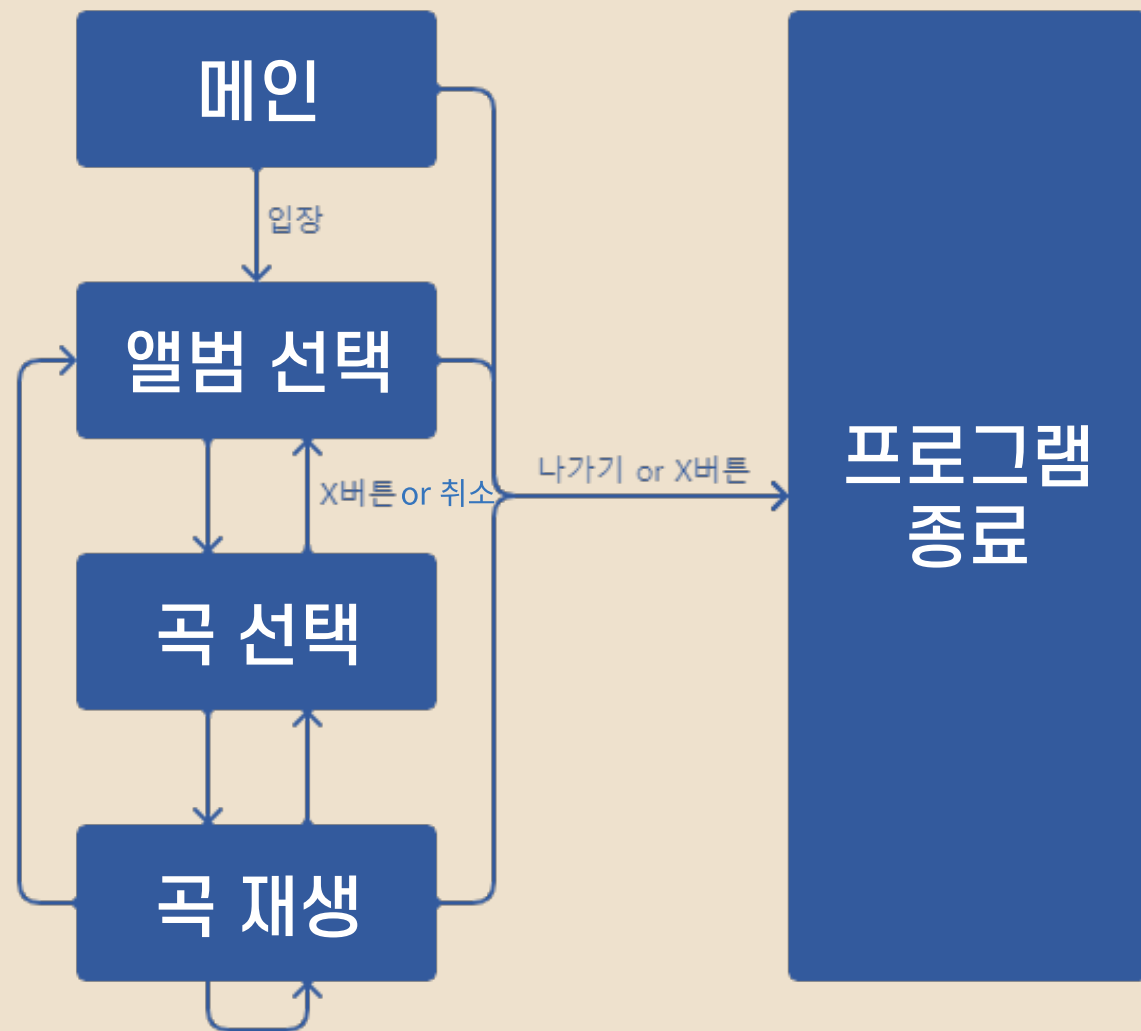
VOL 1



TOY PLAYER

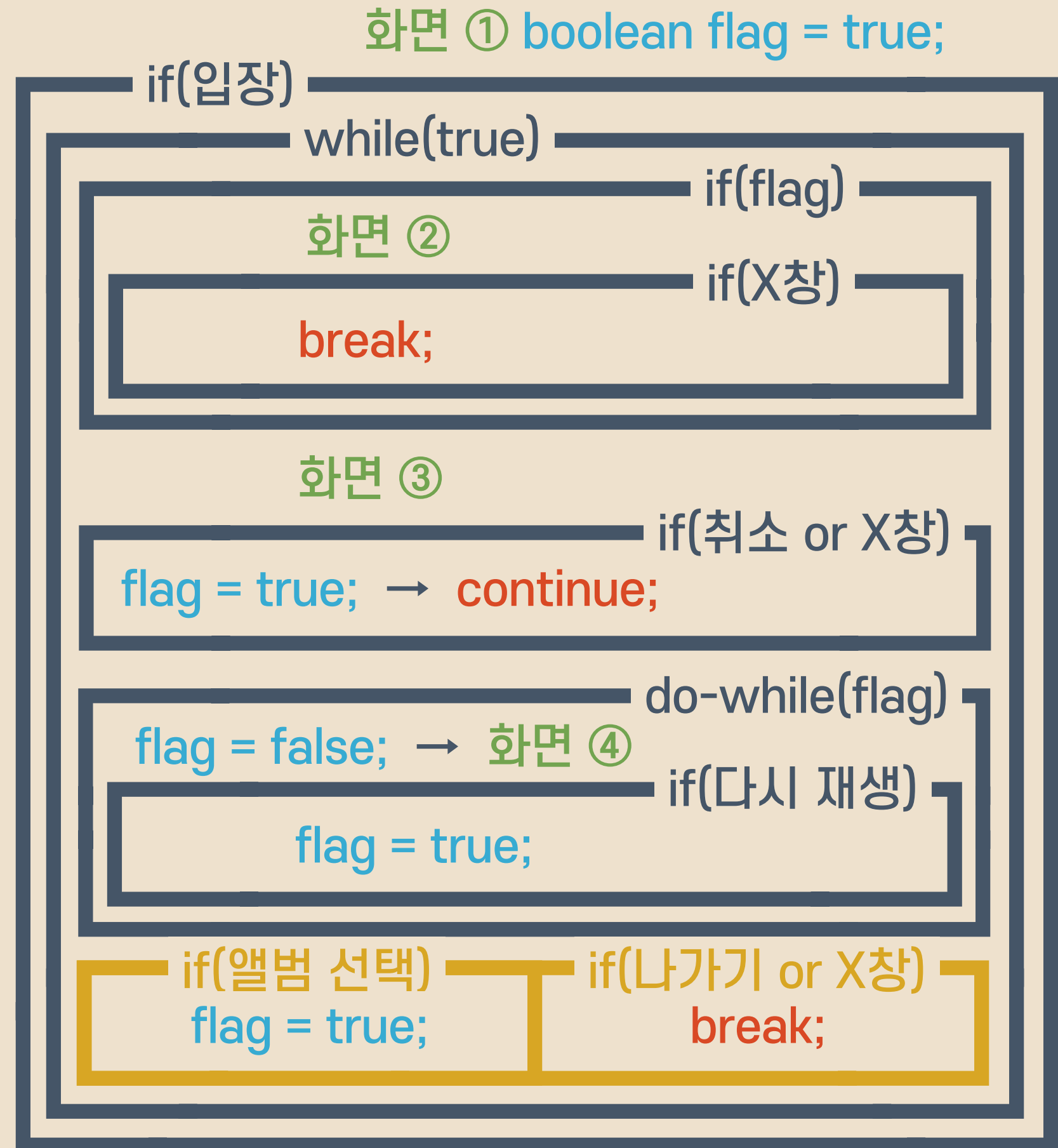
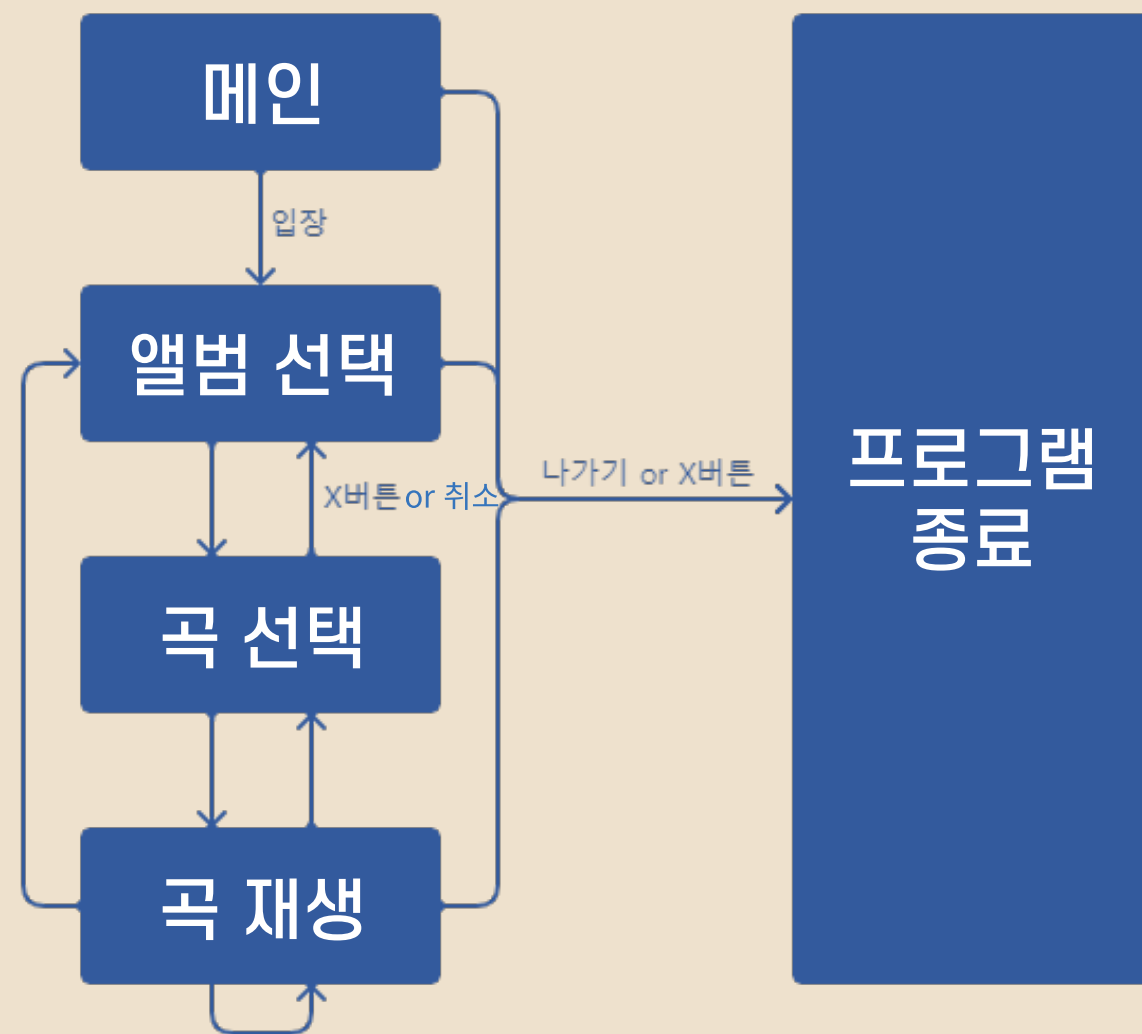
제어문, flag

- flag = true;
(false → true)
- "다시 재생" 선택 시 flag에 의해
do-while문 반복



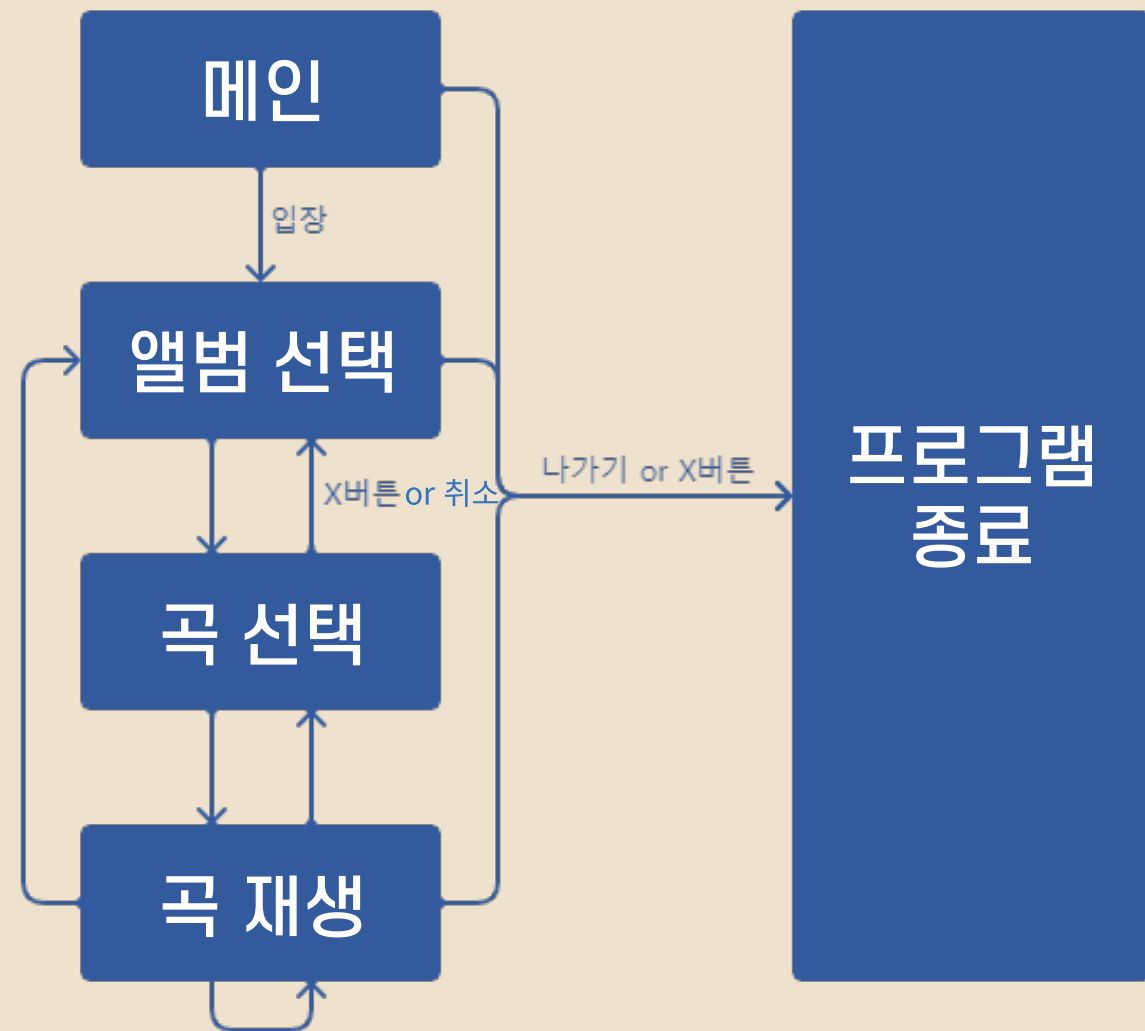
제어문, flag

- `flag = false;`
- "다시 재생" 선택 X
⇒ do-while문 탈출

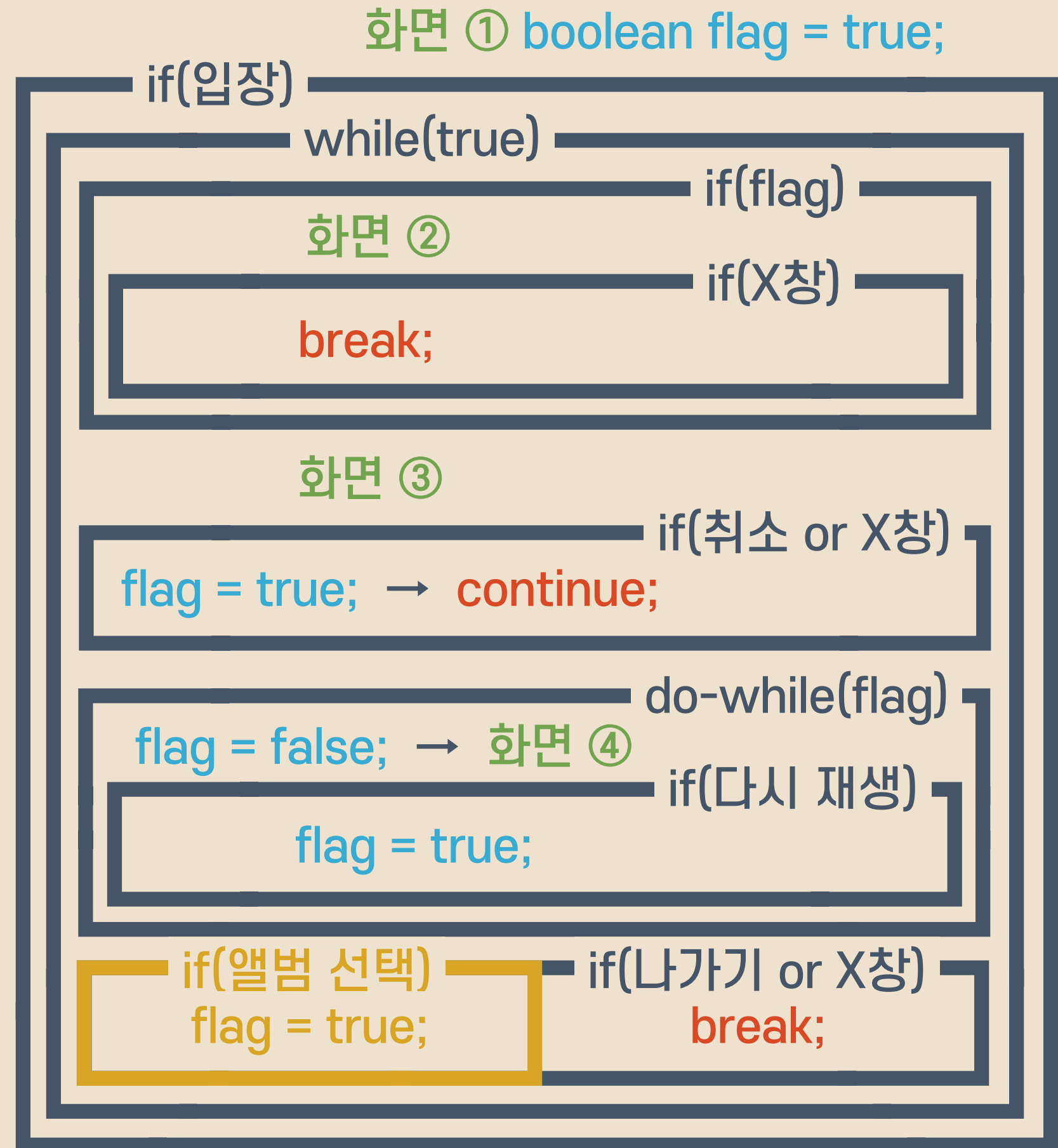


제어문, flag

- `flag = true;`
(`false` → `true`)
- "앨범 선택" 선택 시 flag에 의해
화면 ②로 이동



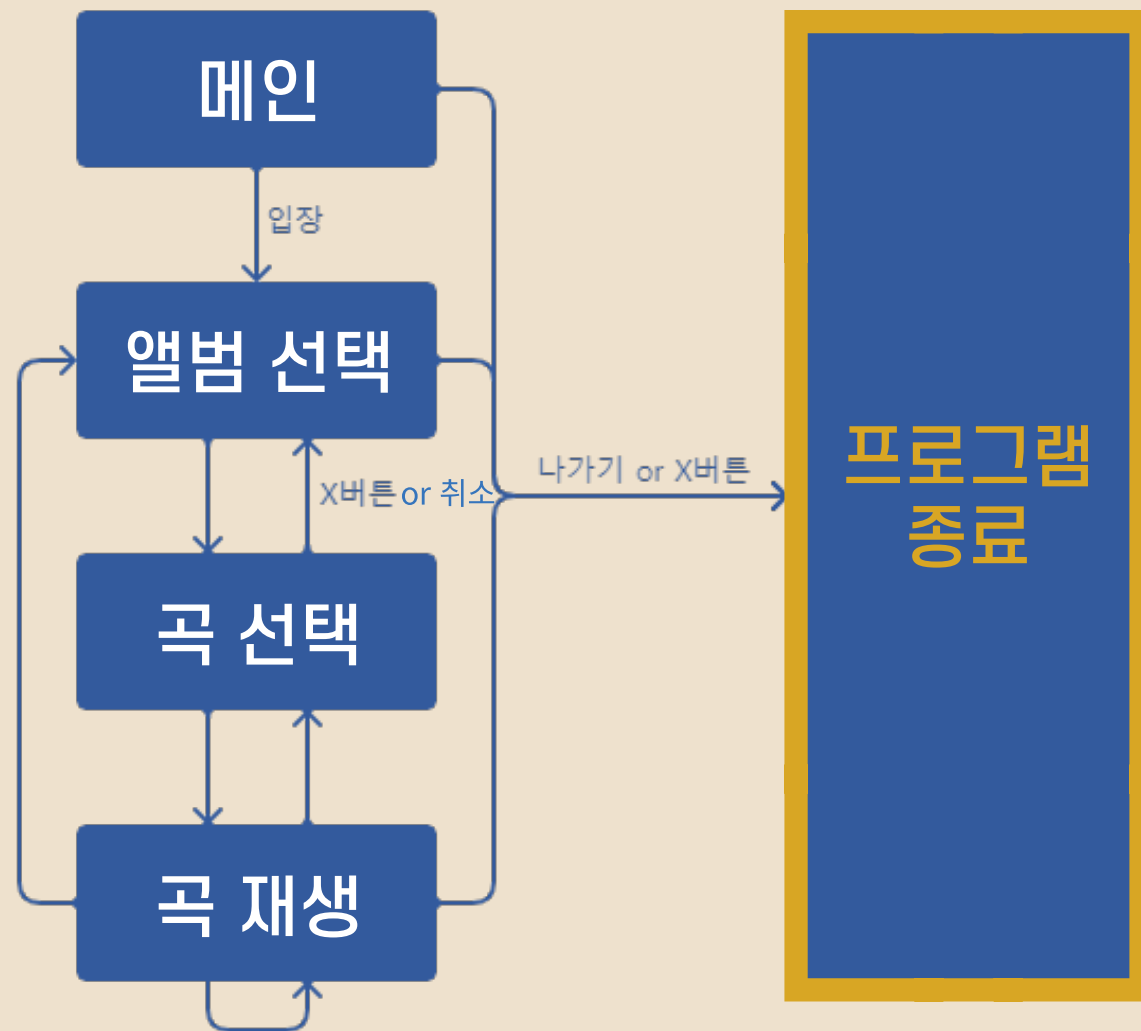
VOL 1



TOY PLAYER

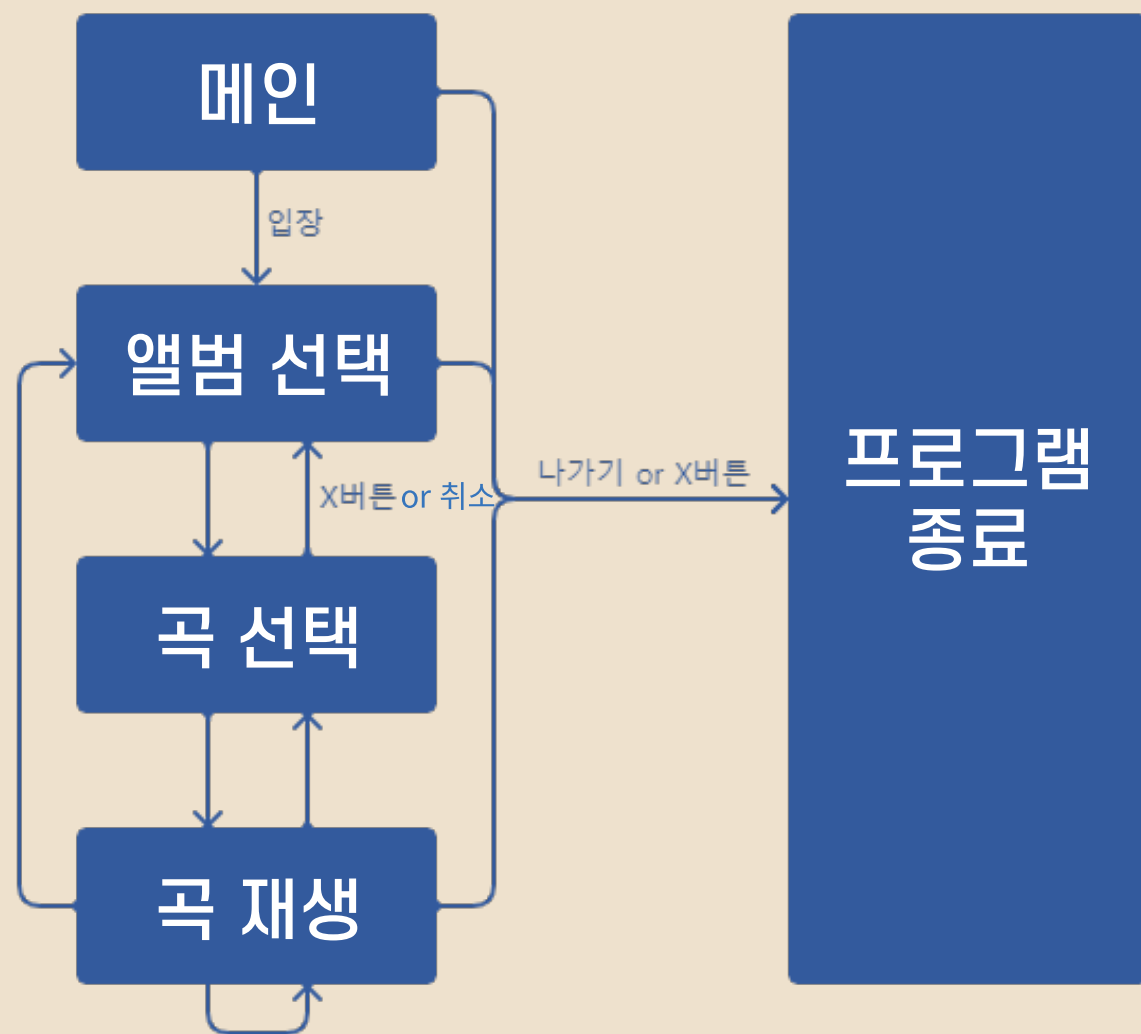
제어문, flag

- flag = false;
- "나가기" or "X창" 선택 시 프로그램 종료



제어문, flag

- `flag = false;`
- "곡 선택" 선택 시 flag에 의해 화면 ③으로 이동



03

시연

04

소감

감사합니다

유세영