**랜덤숫자 생성기 - 기능 중심 파일별 TODO**

**📁 main.dart**

// TODO 108강: 앱 초기화 및 MaterialApp 설정

// [ ] Flutter 앱 진입점 만들기 (runApp 함수)

// [ ] MaterialApp 위젯으로 앱 감싸기

// [ ] HomeScreen을 첫 화면으로 설정

// [ ] import 경로 정확히 설정

//

// 💡 핵심 개념:

// - Flutter 앱의 시작점 역할

// - MaterialApp vs CupertinoApp 차이

// - home 속성의 역할

//

// 📱 구조 이해:

// - 최상위 위젯 트리의 시작점

// - 전체 앱을 감싸는 컨테이너 역할

import 'package:flutter/material.dart';

// TODO: HomeScreen import 추가

void main() {

// TODO: runApp() 함수로 앱 실행

// TODO: MaterialApp(home: HomeScreen()) 구조 구현

}

**📁 constant/color.dart**

// TODO 109강: 앱 전체 색상 상수 관리 시스템

// [ ] const Color 타입으로 색상 상수 정의

// [ ] primaryColor (어두운 배경) 정의

// [ ] redColor (버튼, 강조색) 정의

// [ ] blueColor (보조 색상) 정의

//

// 💡 핵심 개념:

// - const 키워드로 컴파일 타임 상수 생성

// - Color(0xFF헥스코드) 형식 이해

// - 전역 상수의 네이밍 컨벤션

//

// 🔗 다른 파일과 연결:

// - 모든 화면에서 import해서 사용

// - 일관된 디자인 시스템의 기반

import 'package:flutter/material.dart';

// TODO: primaryColor, redColor, blueColor 상수 정의

// 힌트: const Color primaryColor = Color(0xFF헥스코드);

**📁 component/number\_to\_image.dart**

// TODO 109-110강: 숫자를 이미지로 변환하는 재사용 컴포넌트

// [ ] StatelessWidget 클래스 생성

// [ ] required int number 매개변수 받는 생성자 설계

// [ ] 숫자를 문자열로 변환 (toString())

// [ ] 문자열을 한 글자씩 분리 (split(''))

// [ ] map() 함수로 각 글자를 Image.asset으로 변환

// [ ] Row 위젯으로 이미지들을 가로 배치

//

// 💡 핵심 개념:

// - .toString().split('') 메서드 체이닝

// - map() 함수의 동작 원리와 활용

// - .toList()로 Iterable을 List로 변환

// - 'asset/img/$number.png' 경로 동적 생성

//

// 📱 레이아웃 구조:

// - Row 위젯: 가로 배치 컨테이너

// - children 속성: 위젯 리스트 배치

// - Image.asset: 앱 내부 이미지 리소스 로드

//

// 🔗 컴포넌트 설계:

// - 외부에서 number만 전달받는 순수한 컴포넌트

// - 재사용성을 위한 Props 기반 설계

import 'package:flutter/material.dart';

class NumberToImage extends StatelessWidget {

// TODO: final int number 속성 추가

// TODO: required this.number 생성자 구현

@override

Widget build(BuildContext context) {

return Row(

children: [

// TODO: number.toString().split('').map() 체이닝 구현

// TODO: 각 문자를 Image.asset('asset/img/$문자.png')로 변환

// TODO: width: 50.0, height: 70.0 크기 설정

// TODO: .toList()로 최종 변환

],

);

}

}

**📁 screen/home\_screen.dart**

// TODO 111-114강: 메인 화면 - 상태관리와 네비게이션 시스템

// [ ] StatefulWidget으로 변경하여 상태 관리 가능하게 만들기

// [ ] List<int> numbers 상태 변수 선언 (초기값: [123, 456, 789])

// [ ] int maxNumber 상태 변수 선언 (초기값: 1000)

// [ ] generateRandomNumber() 함수 구현 (랜덤 숫자 3개 생성)

// [ ] onSettingIconPressed() 함수 구현 (설정 화면 네비게이션)

// [ ] \_Header, \_Body, \_Footer 위젯 분리 설계

//

// 💡 상태관리 힌트:

// - StatefulWidget vs StatelessWidget 차이점

// - setState() 함수의 역할과 동작 원리

// - 상태 변수가 변경되면 build() 재실행

// - 자식 위젯들에게 상태를 props로 전달하는 패턴

//

// 📱 레이아웃 구조:

// - Scaffold: 기본 화면 구조 (Material Design)

// - SafeArea: 시스템 UI 영역 피하기

// - Column: 세로 배치 (Header + Body + Footer)

// - CrossAxisAlignment.stretch: 가로 전체 너비 사용

// - Expanded: 남는 공간을 Body가 차지하도록

//

// 🔗 컴포넌트 연결:

// - \_Header: onPressed 콜백 전달

// - \_Body: numbers 리스트 전달

// - \_Footer: onPressed 콜백 전달

//

// ⚡ 핵심 로직:

// - Random 클래스로 랜덤 숫자 생성

// - Set<int>으로 중복 제거

// - while 루프로 3개까지 생성

// - Navigator.push()로 화면 전환 및 결과 받기

import 'package:flutter/material.dart';

import 'dart:math';

// TODO: 필요한 import들 추가 (color.dart, number\_to\_image.dart, setting\_screen.dart)

class HomeScreen extends StatefulWidget {

const HomeScreen({super.key});

@override

State<HomeScreen> createState() => \_HomeScreenState();

}

class \_HomeScreenState extends State<HomeScreen> {

// TODO: List<int> numbers 상태 변수 선언 (초기값: [123, 456, 789])

// TODO: int maxNumber 상태 변수 선언 (초기값: 1000)

@override

Widget build(BuildContext context) {

return Scaffold(

backgroundColor: primaryColor,

body: SafeArea(

child: Padding(

padding: EdgeInsets.symmetric(horizontal: 16.0),

child: Column(

crossAxisAlignment: CrossAxisAlignment.stretch,

children: [

// TODO: \_Header 위젯 배치 (onPressed: onSettingIconPressed)

// TODO: \_Body 위젯 배치 (numbers: numbers)

// TODO: \_Footer 위젯 배치 (onPressed: generateRandomNumber)

],

),

),

),

);

}

// TODO: 설정 화면으로 네비게이션하는 함수

onSettingIconPressed() async {

// 힌트: Navigator.of(context).push() 사용

// 힌트: MaterialPageRoute로 SettingScreen 전달

// 힌트: maxNumber를 SettingScreen에 전달

// 힌트: await로 결과 받아서 maxNumber 업데이트

}

// TODO: 랜덤 숫자 3개 생성하는 함수

generateRandomNumber() {

// 힌트: Random() 인스턴스 생성

// 힌트: Set<int> newNumbers = {} 로 중복 제거

// 힌트: while (newNumbers.length < 3) 루프

// 힌트: rand.nextInt(maxNumber)로 랜덤 숫자 생성

// 힌트: setState()로 numbers 업데이트

}

}

// TODO: \_Header 위젯 - 제목과 설정 아이콘

class \_Header extends StatelessWidget {

// TODO: VoidCallback onPressed 받는 생성자

@override

Widget build(BuildContext context) {

// TODO: Row 위젯으로 제목과 IconButton 배치

// 힌트: MainAxisAlignment.spaceBetween 사용

// 힌트: Text('랜덤숫자 생성기') + IconButton(Icons.settings)

}

}

// TODO: \_Body 위젯 - 숫자들 표시 영역

class \_Body extends StatelessWidget {

// TODO: List<int> numbers 받는 생성자

@override

Widget build(BuildContext context) {

// TODO: Expanded + Column으로 숫자들 세로 배치

// 힌트: MainAxisAlignment.center로 중앙 정렬

// 힌트: numbers.map((e) => NumberToImage(number: e)).toList()

}

}

// TODO: \_Footer 위젯 - 생성 버튼

class \_Footer extends StatelessWidget {

// TODO: VoidCallback onPressed 받는 생성자

@override

Widget build(BuildContext context) {

// TODO: ElevatedButton 구현

// 힌트: style로 backgroundColor, foregroundColor 설정

// 힌트: Text('생성하기!') 표시

}

}

**📁 screen/setting\_screen.dart**

// TODO 115-119강: 설정 화면 - 데이터 전달과 Slider 상태관리

// [ ] 이전 화면에서 maxNumber 받는 생성자 구현

// [ ] double maxNumber 상태 변수로 Slider 값 관리

// [ ] initState()에서 받은 int를 double로 변환하여 초기화

// [ ] onSliderChanged() 함수로 Slider 값 변경 처리

// [ ] onSavePressed() 함수로 결과 반환하며 이전 화면으로 이동

// [ ] \_Number, \_Slider, \_Button 위젯 분리 설계

//

// 💡 상태관리 힌트:

// - 생성자로 받은 데이터를 상태로 초기화하는 패턴

// - initState() 생명주기 메서드 활용

// - widget.프로퍼티로 StatefulWidget 속성 접근

// - int ↔ double 타입 변환 (toInt(), toDouble())

//

// 📱 레이아웃 구조:

// - Scaffold + SafeArea + Padding 기본 구조

// - Column으로 \_Number + \_Slider + \_Button 세로 배치

// - CrossAxisAlignment.stretch로 가로 전체 사용

//

// 🔗 컴포넌트 연결:

// - \_Number: maxNumber 값 전달하여 시각적 표시

// - \_Slider: value와 onChanged 콜백 전달

// - \_Button: onPressed 콜백으로 저장 기능 연결

//

// ⚡ 핵심 로직:

// - Slider의 min(1000), max(100000) 범위 설정

// - 실시간으로 변경되는 값을 NumberToImage로 표시

// - Navigator.pop(값)으로 결과 반환

import 'package:flutter/material.dart';

// TODO: 필요한 import들 추가

class SettingScreen extends StatefulWidget {

// TODO: final int maxNumber 받는 생성자 정의

@override

State<SettingScreen> createState() => \_SettingScreenState();

}

class \_SettingScreenState extends State<SettingScreen> {

// TODO: double maxNumber 상태 변수 선언

@override

void initState() {

super.initState();

// TODO: widget.maxNumber.toDouble()로 초기화

}

@override

Widget build(BuildContext context) {

return Scaffold(

backgroundColor: primaryColor,

body: SafeArea(

child: Padding(

padding: EdgeInsets.symmetric(horizontal: 16.0),

child: Column(

crossAxisAlignment: CrossAxisAlignment.stretch,

children: [

// TODO: \_Number 위젯 배치 (maxNumber: maxNumber)

// TODO: \_Slider 위젯 배치 (value: maxNumber, onChanged: onSliderChanged)

// TODO: \_Button 위젯 배치 (onPressed: onSavePressed)

],

),

),

),

);

}

// TODO: 저장 버튼 눌렀을 때 실행되는 함수

onSavePressed() {

// 힌트: Navigator.of(context).pop(maxNumber.toInt())

}

// TODO: 슬라이더 값 변경시 실행되는 함수

onSliderChanged(double value) {

// 힌트: setState(() { maxNumber = value; })

}

}

// TODO: \_Number 위젯 - 현재 설정값을 NumberToImage로 표시

class \_Number extends StatelessWidget {

// TODO: double maxNumber 받는 생성자

@override

Widget build(BuildContext context) {

// TODO: Expanded + Container 조합

// 힌트: NumberToImage(number: maxNumber.toInt()) 사용

// 힌트: Container는 중앙 정렬을 위한 래퍼

}

}

// TODO: \_Slider 위젯 - 값 조정용 슬라이더

class \_Slider extends StatelessWidget {

// TODO: double value와 ValueChanged<double> onChanged 받는 생성자

@override

Widget build(BuildContext context) {

// TODO: Slider 위젯 구현

// 힌트: value, min: 1000, max: 100000, onChanged 속성

// 힌트: activeColor: redColor로 스타일 적용

}

}

// TODO: \_Button 위젯 - 저장 버튼

class \_Button extends StatelessWidget {

// TODO: VoidCallback onPressed 받는 생성자

@override

Widget build(BuildContext context) {

// TODO: ElevatedButton 구현

// 힌트: style로 backgroundColor, foregroundColor 설정

// 힌트: Text('저장!') 표시

}

}

**🎯 학습 포인트 요약**

**📚 강의별 핵심 개념**

* **108강**: Flutter 앱 구조, MaterialApp
* **109강**: Image.asset, 동적 경로 생성
* **110강**: String 메서드 체이닝, map() 함수
* **111강**: State Lifting, StatefulWidget
* **112강**: Random 클래스, Set 자료구조
* **113강**: 자료구조 이론 (Queue/Stack)
* **114강**: Navigator, 화면 전환
* **115강**: 화면간 데이터 전달
* **116강**: 컴포넌트 재사용성
* **117강**: Slider 위젯, ValueChanged 콜백
* **118강**: Navigator.pop() 데이터 반환
* **119강**: 생성자를 통한 데이터 전달

**🔧 상태관리 패턴**

* **Local State**: 각 화면의 상태 (numbers, maxNumber)
* **Props 전달**: 부모 → 자식 데이터 흐름
* **Callback 전달**: 자식 → 부모 이벤트 흐름
* **State Lifting**: 공유 상태를 상위로 이동

**📱 레이아웃 구조 패턴**

* **Scaffold**: 기본 화면 구조
* **Column**: 세로 배치 컨테이너
* **Row**: 가로 배치 컨테이너
* **Expanded**: 유동적 공간 분배
* **SafeArea**: 시스템 UI 영역 처리

**🚀 롤팡 연결 포인트**

* **numbers 리스트** → **messages 리스트**
* **NumberToImage** → **MessageWidget**
* **Navigator 패턴** → **화면 전환 시스템**
* **Slider 상태관리** → **감정 설정 시스템**
* **컴포넌트 분리** → **재사용 위젯 설계**

**📅 추천 학습 순서**

1. **1일차**: main.dart + color.dart (기초 설정) - 1시간
2. **2일차**: number\_to\_image.dart (컴포넌트 이해) - 1.5시간
3. **3일차**: home\_screen.dart 기본 구조 (상태관리 기초) - 2시간
4. **4일차**: home\_screen.dart 고급 기능 (랜덤 생성, 네비게이션) - 2시간
5. **5일차**: setting\_screen.dart (화면간 통신) - 1.5시간

**총 8시간으로 Flutter 핵심 패턴 완전 체화! 🎯**