**스마트폰 응용 프로그래밍**

**<2주차 과제: 퀴즈 앱 만들기>**

**20194111 최민규**

|  |
| --- |
| 과제 내용 |
| 퀴즈 문제 10문제를 차례대로 보여주고, 정답을 검사하는 기능을 가진 앱을 작성하기. |

|  |  |
| --- | --- |
| 프로젝트 명 | 20194111Quiz |
| 프로젝트  설명 | 넌센스 퀴즈 10문제를 푸는 앱을 작성해보자.  문제는 모두 객관식으로, 보기 4개를 포함하고 있다.  보기 중 원하는 답을 고르고, ‘제출’ 버튼을 통해 정답을 입력한다.  정답을 검사하고 오답/정답을 체크한다.  마지막 문제까지 모두 풀고나면 본인이 맞힌 답의 개수를 출력한다.  **\*수업시간에 배운 findViewByID를 통해 구현하려다, 서칭을 통해 ‘뷰 바인드(View Bind) 기능에 대해 찾을 수 있었고, 이를 통해 프로그램을 구현하였습니다.**  참고 페이지:  <https://developer.android.com/topic/libraries/view-binding?hl=ko>  <https://aries574.tistory.com/category/%EC%95%88%EB%93%9C%EB%A1%9C%EC%9D%B4%EB%93%9C> |

**1. gradle에서 바인딩 활성화 build.gradle.kts**

|  |
| --- |
| 코드 |
| //바인딩 설정: findViewbuId 없이 뷰에 바로 접근 buildFeatures **{** viewBinding = true **}** |

**2. 문제 풀이 화면 레이아웃 구성 activity\_main.xml**

|  |  |
| --- | --- |
| 코드 | 디자인 |
| <?xml version="1.0" encoding="utf-8"?> <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:app="http://schemas.android.com/apk/res-auto"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:gravity="center"  android:orientation="vertical"  android:padding="10dp"  tools:context=".MainActivity">   <TextView  android:id="@+id/question\_text"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:gravity="center"  android:text="@string/question"  android:textSize="18sp"  android:textStyle="bold" />   <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginTop="20dp"  android:gravity="center\_vertical"  android:orientation="horizontal">   <ProgressBar  android:id="@+id/progressBar"  style="?android:progressBarStyleHorizontal"  android:layout\_width="0dp"  android:layout\_height="50dp"  android:layout\_weight="1"  android:max="5"  android:progress="0" />   <TextView  android:id="@+id/progress\_text"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:padding="10dp"  android:text="@string/count\_label"  android:textColor="@color/black"  android:textSize="20sp" />   </LinearLayout>   <TextView  android:id="@+id/option1\_text"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginTop="20dp"  android:background="@drawable/option\_background"  android:gravity="center"  android:padding="10dp"  android:text="답변1"  android:textSize="20sp" />   <TextView  android:id="@+id/option2\_text"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginTop="20dp"  android:background="@drawable/option\_background"  android:gravity="center"  android:padding="10dp"  android:text="답변2"  android:textSize="20sp" />   <TextView  android:id="@+id/option3\_text"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginTop="20dp"  android:background="@drawable/option\_background"  android:gravity="center"  android:padding="10dp"  android:text="답변3"  android:textSize="20sp" />   <TextView  android:id="@+id/option4\_text"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginTop="20dp"  android:background="@drawable/option\_background"  android:gravity="center"  android:padding="10dp"  android:text="답변4"  android:textSize="20sp" />   <Button  android:id="@+id/submit\_btn"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginTop="20dp"  android:text="@string/submit"  android:textSize="20sp"  android:textStyle="bold" />    </LinearLayout> | \*progressBar를 통해 현재까지의 진행상황을 표시하고, 답변 네 개 중 하나를 선택할 수 있도록 구성했습니다.  \*아래 제출 버튼을 통해 [제출]/[다음]의 기능을 수행합니다. |

**3. 사용할 문자열 파일 정리 strings.xml**

|  |
| --- |
| 코드 |
| <resources>  <string name="app\_name">20194111Quiz</string>   <string name="question">질문</string>  <string name="count\_label">%d/%d</string>  <string name="submit">%s</string>  </resources> |

**\*뷰 바인딩 기능을 통해 위의 %d, %s에 각각 해당하는 내용을 동적으로 출력할 수 있도록 구성하였습니다.**

**4. 활용할 디자인 파일 (.xml) 작성**

|  |  |
| --- | --- |
|  | \*누르기 전 버튼, 누른 후 버튼을 각각 하얀색/보라색으로 표시할 수 있도록 xml 파일 작성  \*정답인 경우 버튼을 초록색으로,  오답인 경우 누른 버튼을 빨간색으로 표시하도록 각각의 xml 파일 작성 |

**5. 퀴즈 종료 후 결과창 레이아웃 작성 activity\_result.xml**

|  |  |
| --- | --- |
| 코드 | 디자인 |
| <?xml version="1.0" encoding="utf-8"?> <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:app="http://schemas.android.com/apk/res-auto"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:gravity="center"  android:orientation="vertical"  tools:context=".ResultActivity"  >   <TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="결과"  android:textSize="30sp"  android:textStyle="bold"/>   <TextView  android:id="@+id/score\_text"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:textSize="20sp"  android:textStyle="bold"/>   <Button  android:id="@+id/reset\_btn"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="다시 하기"  android:textSize="20sp"/>  </LinearLayout> | \*결과창에 맞힌 문제 수 / 전체 문제 수 표시  \*[다시하기] 버튼을 통해 처음으로 돌릴 수 있도록 작성. |

**6.퀴즈를 클래스 정의 QuestinData.kt**

|  |
| --- |
| 코드 |
| package com.example.a20194111quiz  data class Question(  var id: Int,  var question:String,  var option\_one:String,  var option\_two: String,  var option\_three: String,  var option\_four: String,  var correct\_answer: Int ) |

각 퀴즈들은 id와 문제, 4지선다와 정답값을 가진 클래스로 정의한다.

**7. 퀴즈 데이터 클래스 정의 QuestionData.kt**

|  |
| --- |
| 코드 |
| package com.example.a20194111quiz  object QuestionData {   fun getQuestion(): ArrayList<Question>{   val queList: ArrayList<Question> = *arrayListOf*()     val q1 = Question(  1,  "못팔고도 돈 번 사람은?",  "철물점 주인",  "빵집 주인",  "슈퍼마켓 주인",  "편의점 주인",  1  )   val q2 = Question(  1,  "도둑이 가장 싫어하는 아이스크림은? ",  "바밤바",  "스크류바",  "비비빅",  "누가바",  4  )   val q3 = Question(  1,  "한국에서 가장 급하게 만든 떡은?",  "가래떡",  "쑥떡쑥떡",  "헐레벌떡",  "시루떡",  3  )   val q4 = Question(  1,  "호주의 돈을 뭐라고 하는가?",  "산와머니",  "호주머니",  "쇼미더머니",  "오스트레일리아",  2  )   val q5 = Question(  1,  "학생들이 싫어하는 피자는?",  "하와이안 피자",  "포테이토 피자",  "책 피자",  "페퍼로니 피자",  3  )   val q6 = Question(  1,  " 도둑이 가장 좋아하는 아이스크림은? ",  "보석바",  "누가바",  "바밤바",  "비비빅",  1  )   val q7 = Question(  1,  "바나나가 웃으면?",  "나한테 반하나?",  "바나나킥",  "바나나는 원래 하얗다",  "바나나맛우유",  2  )   val q8 = Question(  1,  "왕이 넘어지면?",  "킹덤",  "왕릉",  "킹콩",  "왕관",  3  )   val q9 = Question(  1,  "새우가 출연하는 드라마는?",  "대하드라마",  "금토드라마",  "일일드라마",  "사극드라마",  1  )   val q10 = Question(  1,  "해가 우는 곳은? ",  "해남",  "해녀촌",  "해삼",  "해운대",  4  )   queList.add(q1)  queList.add(q2)  queList.add(q3)  queList.add(q4)  queList.add(q5)  queList.add(q6)  queList.add(q7)  queList.add(q8)  queList.add(q9)  queList.add(q10)   return queList  } } |

총 10개의 Question을 작성하고 queList에 추가한다.

**8. 퀴즈 종료 후 결과창 액티비티 작성ResultActivity.kt**

|  |
| --- |
| 코드 |
| package com.example.a20194111quiz  import android.content.Intent import androidx.appcompat.app.AppCompatActivity import android.os.Bundle import com.example.a20194111quiz.databinding.ActivityMainBinding import com.example.a20194111quiz.databinding.ActivityResultBinding  class ResultActivity : AppCompatActivity() {   private lateinit var binding: ActivityResultBinding   override fun onCreate(savedInstanceState: Bundle?) {  super.onCreate(savedInstanceState)  binding = ActivityResultBinding.inflate(*layoutInflater*)  setContentView(binding.*root*)   val score = *intent*.getIntExtra("score",0)  val totalSize = *intent*.getIntExtra("totalSize",0)   //점수 보여주기  binding.scoreText.*text* = getString(R.string.*count\_label*, score, totalSize)   //다시하기 버튼  binding.resetBtn.setOnClickListener**{** val intent = Intent(this@ResultActivity, MainActivity::class.*java*)  startActivity(intent)  **}** } } |

\*문제를 다 풀고나면 정답 점수를 보여주고, 다시하기 버튼을 통해 초기화면으로 돌아갈 수 있도록 한다.

**9.메인 액티비티 작성 MainActivity.kt**

|  |
| --- |
| 코드 |
| package com.example.a20194111quiz  import android.content.Intent import android.graphics.Color import android.graphics.Typeface import androidx.appcompat.app.AppCompatActivity import android.os.Bundle import android.renderscript.ScriptGroup.Binding import android.view.View import android.widget.TextView import androidx.appcompat.app.AlertDialog import androidx.core.content.ContextCompat import com.example.a20194111quiz.databinding.ActivityMainBinding  class MainActivity : AppCompatActivity(), View.OnClickListener {   private lateinit var binding : ActivityMainBinding   private var currentPosition: Int = 1  private var selectedOption: Int = 0  private var score: Int = 0   private lateinit var questionList: ArrayList<Question>   override fun onCreate(savedInstanceState: Bundle?) {  super.onCreate(savedInstanceState)  binding = ActivityMainBinding.inflate(*layoutInflater*)  val view = binding.*root* setContentView(binding.*root*)   //질문 리스트 가져오기  questionList = QuestionData.getQuestion()   //화면 셋팅  getQuestionData()   binding.option1Text.setOnClickListener (this)  binding.option2Text.setOnClickListener (this)  binding.option3Text.setOnClickListener (this)  binding.option4Text.setOnClickListener (this)    //답변 체크 이벤트  binding.submitBtn.setOnClickListener**{** if (selectedOption != 0){  val question = questionList[currentPosition-1]   //정답 체크  if (selectedOption != question.correct\_answer){ //오답  setColor(selectedOption, R.drawable.*wrong\_option\_background*)   callDialog("오답","정답 ${question.correct\_answer}")  }else{  score++  }  setColor(question.correct\_answer, R.drawable.*correct\_option\_background*)   if (currentPosition == questionList.size){  binding.submitBtn.*text* = getString(R.string.*submit*, "끝")  }else{  binding.submitBtn.*text* = getString(R.string.*submit*,"다음")  }  }   else{  //위치값 상승  currentPosition++  when{  currentPosition <= questionList.size ->{  //다음 문제 셋팅  getQuestionData()  }   else ->{  //결과 액티비티로  val intent = Intent(this@MainActivity,ResultActivity::class.*java*)  intent.putExtra("score",score)  intent.putExtra("totalSize",questionList.size)  startActivity(intent)  finish()  }  }  }  //선택값 초기화  selectedOption = 0  **}** //submitBtn   } //onCreate   //답변 배경 색상 변경  private fun setColor(opt:Int, color:Int){  when(opt){  1 -> binding.option1Text.*background* = ContextCompat.getDrawable(this,color)  2 -> binding.option2Text.*background* = ContextCompat.getDrawable(this,color)  3 -> binding.option3Text.*background* = ContextCompat.getDrawable(this,color)  4 -> binding.option4Text.*background* = ContextCompat.getDrawable(this,color)  }  }   //문제 셋팅  private fun getQuestionData(){   //답변 설정 초기화  setOptionStyle()   //질문 변수에 담기  val question = questionList[currentPosition-1]   //상태바 위치  binding.progressBar.*progress* = currentPosition   //상태바 최대값  binding.progressBar.*max* = questionList.size   //현재 위치 표시  binding.progressText.*text* = getString(R.string.*count\_label*, currentPosition, questionList.size)   //질문 표시  binding.questionText.*text* = question.question   //답변 표시  binding.option1Text.*text* = question.option\_one  binding.option2Text.*text* = question.option\_two  binding.option3Text.*text* = question.option\_three  binding.option4Text.*text* = question.option\_four   setSubmitBtn("제출")  }   //정답 확인 다이얼로그  private fun callDialog(alertTitle:String, correctAnswer: String){   AlertDialog.Builder(this)  .setTitle(alertTitle)  .setMessage("정답: $correctAnswer")  .setPositiveButton("OK")**{** dialogInterface, i**->** dialogInterface.dismiss() //창 닫기  **}** .setCancelable(false)  .show()  }    //제출 버튼 텍스트 설정  private fun setSubmitBtn(name:String)  {  binding.submitBtn.*text* = getString(R.string.*submit*, name)  }   //답변 스타일 설정  private fun setOptionStyle(){  var optionList: ArrayList<TextView> = *arrayListOf*()  optionList.add(binding.option1Text)  optionList.add(binding.option2Text)  optionList.add(binding.option3Text)  optionList.add(binding.option4Text)   //답변 텍스트 뷰 설정  for (op in optionList){  op.setTextColor(Color.parseColor("#555151"))  op.*background* = ContextCompat.getDrawable(this, R.drawable.*option\_background*)  op.*typeface* = Typeface.*DEFAULT* }  }   //답변 선택 이벤트  private fun selectedOptionStyle(view: TextView, opt: Int){   //옵션 초기화  setOptionStyle()   //위치 담기  selectedOption = opt  view.setTextColor((Color.parseColor("#5F00FF")))  view.*background* = ContextCompat.getDrawable(this,R.drawable.*selected\_option\_background*)  view.*typeface* = Typeface.*DEFAULT\_BOLD* }   //클릭 이벤트 재정의  override fun onClick(view: View){  when (view.*id*){  R.id.*option1\_text* -> selectedOptionStyle(binding.option1Text, 1)  R.id.*option2\_text* -> selectedOptionStyle(binding.option2Text, 2)  R.id.*option3\_text* -> selectedOptionStyle(binding.option3Text, 3)  R.id.*option4\_text* -> selectedOptionStyle(binding.option4Text, 4)  }  } } |

**\*실행결과**

|  |
| --- |
| 문제화면 - TextView를 통해 문제가 표시되고,  각 버튼에 네 개의 선지가 출력된다.  누른 버튼은 보라색 텍스트와 외곽선으로 표시된다. |
| 정답을 고르고 [제출]버튼을 누른 경우 정답에 해당하는 버튼이 초록색으로 표시된다. |
| 반대로 오답을 고르고 [제출] 버튼을 누른 경우, 해당 문제의 정답에 대한 메시지가 출력되며 오답 버튼은 빨간색, 정답 버튼은 초록색으로 표시된다. |
| 문제를 모두 풀고나면 결과창에 맞힌 문제 수/ 전체 문제 수가 출력된다.  [다시 하기] 버튼을 통해 첫 화면으로 돌아갈 수 있다. |

|  |
| --- |
| 결론 / 느낀 점 |
| 이번 과제 수행을 통해 LinearLayout을 이용, 수평 방향으로 레이아웃을 작성하는 연습이 되었다.  학기 과제로 작성하려고 했던 문제 풀이 앱을 만드는데 큰 도움이 될 것 같다.  findViewById를 통해 변수에 하나하나 접근하려하다 뷰 바인딩 기능을 통해 코드가 훨씬 깔끔하게 정리되는 것을 느꼈다. |

-