|  |
| --- |
| **학번 : 202114226, 이름 : 송재현** |
| **2024년 12주차 실습 체크 리스트** |
| |  |  | | --- | --- | | **실습 1** | **CRUD** | |
| **실습 1 : CRUD** |
| **소스코드** |
| -Item.kt-  package com.example.week12c.roomDB  import androidx.room.ColumnInfo import androidx.room.Entity import androidx.room.PrimaryKey  @Entity(tableName = "Items") data class Item(  @ColumnInfo(name = "") var itemName: String,  @ColumnInfo(name = "") var itemQuantity: Int,  @PrimaryKey(autoGenerate = true) val itemId: Int = 0 )  -ItemDao.kt-  package com.example.week12c.roomDB  import androidx.room.Dao import androidx.room.Delete import androidx.room.Insert import androidx.room.Query import androidx.room.Update import kotlinx.coroutines.flow.Flow  @Dao interface ItemDao {   @Insert  suspend fun InsertItem(item: Item)   @Update  suspend fun UpdateItem(item: Item)   @Delete  suspend fun DeleteItem(item: Item)   @Query("select \* from items")  fun getAllItems(): Flow<List<Item>> }  -ItemDataBase.kt-  package com.example.week12c.roomDB  import android.content.Context import androidx.room.Database import androidx.room.Room import androidx.room.RoomDatabase  @Database(entities = [Item::class], version = 1, exportSchema = false) abstract class ItemDatabase : RoomDatabase() {  abstract fun getDao(): ItemDao   companion object {  private var database: ItemDatabase? = null   fun getDataBase(context: Context) = database ?: Room.databaseBuilder(  context,  ItemDatabase::class.*java*, "itemdb"  )  .build().*also* **{** database = **it  }** } }  -InputScreen.kt-  package com.example.week12c.screen  import androidx.compose.foundation.layout.Column import androidx.compose.foundation.layout.Row import androidx.compose.foundation.layout.fillMaxWidth import androidx.compose.material3.Button import androidx.compose.material3.Text import androidx.compose.material3.TextField import androidx.compose.runtime.Composable import androidx.compose.runtime.LaunchedEffect import androidx.compose.runtime.getValue import androidx.compose.runtime.mutableStateOf import androidx.compose.runtime.remember import androidx.compose.runtime.setValue import androidx.compose.ui.Modifier import com.example.week12c.roomDB.Item import com.example.week12c.viewmodel.ItemViewModel   @Composable fun InputScreen(viewModel: ItemViewModel, selectedItem: Item? = null) {   var itemId by remember **{** *mutableStateOf*("")  **}** var itemName by remember **{** *mutableStateOf*("")  **}** var itemQuantity by remember **{** *mutableStateOf*("")  **}** val quantity = itemQuantity.*toIntOrNull*() ?: 0  val id = itemId.*toIntOrNull*() ?: 0  val item = Item(itemName, quantity, id)   LaunchedEffect(selectedItem) **{** if (selectedItem != null) {  itemName = selectedItem.itemName  itemQuantity = selectedItem.itemQuantity.toString()  itemId = selectedItem.itemId.toString()  }  **}** fun clearText() {  itemName = ""  itemQuantity = ""  itemId = ""  }   Column(modifier = Modifier.*fillMaxWidth*()) **{** TextField(  value = itemId,  onValueChange = **{** itemId = **it }**,  label = **{** Text("Item ID") **}**,  modifier = Modifier.*fillMaxWidth*()  )  TextField(  value = itemName,  onValueChange = **{** itemName = **it }**,  label = **{** Text("Item Name") **}**,  modifier = Modifier.*fillMaxWidth*()  )  TextField(  value = itemQuantity,  onValueChange = **{** itemQuantity = **it }**,  label = **{** Text("Item Quantity") **}**,  modifier = Modifier.*fillMaxWidth*()  )  Row **{** Button(onClick = **{ }**) **{** Text("Insert")  **}** Button(onClick = **{ }**) **{** Text("Update")  **}** Button(onClick = **{ }**) **{** Text("delete")  **}** Button(onClick = **{ }**) **{** Text("find")  **}  }  }** }  -LazyUI.kt-  package com.example.week12c.screen  import androidx.compose.foundation.clickable import androidx.compose.foundation.layout.Column import androidx.compose.foundation.layout.fillMaxWidth import androidx.compose.foundation.layout.padding import androidx.compose.foundation.lazy.LazyColumn import androidx.compose.foundation.lazy.items import androidx.compose.material3.Divider import androidx.compose.material3.Text import androidx.compose.runtime.Composable import androidx.compose.ui.Modifier import androidx.compose.ui.graphics.Color import androidx.compose.ui.unit.dp import androidx.compose.ui.unit.sp import com.example.week12c.roomDB.Item  @Composable fun ItemList(list: List<Item>, onClick: (Item) -> Unit) {  LazyColumn **{** *items*(list) **{** item **->** ItemUI(item, onClick)  Divider(color = Color.Black, thickness = 1.*dp*)  **}  }** }  @Composable fun ItemUI(item: Item, onClick: (item: Item) -> Unit) {  Column(modifier = Modifier  .*padding*(10.*dp*)  .*fillMaxWidth*()  .*clickable* **{** onClick(item) **}**) **{** Text(item.itemId.toString(), fontSize = 15.*sp*)  Text(item.itemName, fontSize = 15.*sp*)  Text(item.itemQuantity.toString(), fontSize = 15.*sp*)  **}** }  -ItemViewModel.kt-  package com.example.week12c.viewmodel  import androidx.lifecycle.ViewModel import androidx.lifecycle.ViewModelProvider import androidx.lifecycle.*viewModelScope* import com.example.week12c.roomDB.Item import kotlinx.coroutines.flow.MutableStateFlow import kotlinx.coroutines.flow.asStateFlow import kotlinx.coroutines.launch  class ItemViewModelFactory(private val repository: Repository) : ViewModelProvider.Factory {  override fun <T : ViewModel> create(modelClass: Class<T>): T {  if (modelClass.isAssignableFrom(ItemViewModel::class.*java*)) {  return ItemViewModel(repository) as T  }  throw IllegalArgumentException("Unknown ViewModel class")  } }  class ItemViewModel(private val repository: Repository) : ViewModel() {   private var \_itemList = *MutableStateFlow*<List<Item>>(*emptyList*())  val itemList = \_itemList.*asStateFlow*()   fun InsetItem(item: Item) {  *viewModelScope*.*launch* **{** repository.InsertItem(item)  **}** }   fun UpdateItem(item: Item) {  *viewModelScope*.*launch* **{** repository.UpdateItem(item)  **}** }   fun DeleteItem(item: Item) {  *viewModelScope*.*launch* **{** repository.DeleteItem(item)  **}** }   fun getAllItems() {  *viewModelScope*.*launch* **{** repository.getAllItems().collect **{** \_itemList.value = **it  }  }** } }  -Repository.kt-  package com.example.week12c.viewmodel  import com.example.week12c.roomDB.Item import com.example.week12c.roomDB.ItemDatabase  class Repository(private val db: ItemDatabase) {  val dao = db.getDao()   suspend fun InsertItem(item: Item) {  dao.InsertItem(item)  }   suspend fun UpdateItem(item: Item) {  dao.UpdateItem(item)  }   suspend fun DeleteItem(item: Item) {  dao.DeleteItem(item)  }   fun getAllItems() = dao.getAllItems() } |