FilelistActivity

```
package com.android.memo_file

class FilelistActivity : AppCompatActivity() {


    lateinit var binding: ActivityFilelistBinding
    lateinit var rootSD: String
    lateinit var fileDir: File
    lateinit var fileList: ArrayList<String>

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        Log.d("myCheck", "FilelistActivity의 onCreate 메서드 실행")


        binding = ActivityFilelistBinding.inflate(layoutInflater)
        setContentView(binding.root)

        rootSD                                                                                  =
Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_DOCUMENTS).toString()

        fileDir = File(rootSD)
        fileList = ArrayList<String>()

        var list = fileDir.listFiles()

        for(i in 0 until list.size) {
            if(list[i].name != "profileInstalled" || !list[i].isDirectory )  {
                fileList.add(list[i].name.toString())
            }
        }


        binding.listView.adapter = ArrayAdapter<String>(this, android.R.layout.simple_list_item_1,
fileList )


        binding.listView.setOnItemClickListener() { adapterView, view, i, l ->

            var returnIntent = Intent()
            returnIntent.putExtra("fileNameToOpen", fileList[i])
            setResult(RESULT_OK, returnIntent)

            finish()
        }



    }
}
```

Filename 프래그먼트

```
package com.android.memo_file

class Filename : Fragment() {

    private var _binding: FragmentFilenameBinding? = null
    private val binding get() = _binding!!

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        _binding = FragmentFilenameBinding.inflate(inflater, container, false)
        return binding.root
    }

    override fun onDestroyView() {
        super.onDestroyView()
        _binding = null
    }

    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        super.onViewCreated(view, savedInstanceState)

        MyApplication.preferences.setString("MyKey", "FilenameActivity에서 값을 변경하다.")
        val temp = MyApplication.preferences.getString("MyKey", "")
        Log.d("myCheck", "SharedPreference에서 MyKey의 값은 ${temp}")


        binding.button4.setOnClickListener {
            val fileNameToSave = binding.editText2.text.toString()
            parentFragmentManager.setFragmentResult("fileNameRequestKey", Bundle().apply {
                putString("fileNameToSave", fileNameToSave)
            })
            parentFragmentManager.popBackStack()
        }
    }

    companion object {
        @JvmStatic
        fun newInstance() = Filename()
    }
}
```

MainActivity

```
package com.android.memo_file

class MainActivity : AppCompatActivity() {

    lateinit var binding: ActivityMainBinding
    var isSaved = false
    var fileNameToSave = "New"
    var fileNameToOpen = ""
    var state = 1

    lateinit var filenameActivityResult: ActivityResultLauncher<Intent>
    lateinit var filelistActivityResult: ActivityResultLauncher<Intent>

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)

        MyApplication.preferences.setString("MyKey", "MainActivity에서 넣은 값이다.")
        val temp = MyApplication.preferences.getString("MyKey", "")
        Log.d("myCheck", "SharedPreference에서 MyKey의 값은 ${temp}")

        Log.d("myCheck", "MainActivity의 onCreate 메서드 실행")

        state = 1

        binding.button2.isEnabled = false
        binding.textView2.text = fileNameToSave  // 시작할 때는 New

        filelistActivityResult = registerForActivityResult(ActivityResultContracts.StartActivityForResult()) {
            if (it.resultCode == RESULT_OK) {
                fileNameToOpen = it.data?.getStringExtra("fileNameToOpen") ?: ""
                binding.textView2.text = fileNameToOpen
                openSelectedFile()
            }
        }

        binding.button.setOnClickListener {
            if (!isSaved && binding.editTextText.text.isNotEmpty()) {
                if (fileNameToSave == "New") {
                    showFilenameFragment()
                } else {
                    saveToFile()
                    binding.editTextText.setText("")
                    fileNameToSave = "New"
                    binding.textView2.text = fileNameToSave
                }
            } else {
                binding.editTextText.setText("")
                fileNameToSave = "New"
                binding.textView2.text = fileNameToSave
            }
            state = 2
            Log.d("myCheck", "MainActivity에서 '새 메모' 버튼 클릭")
        }
```

```
        binding.button2.setOnClickListener {
            if (fileNameToSave == "New") {
                showFilenameFragment()
            } else {
                saveToFile()
            }
            state = 4
            binding.button2.isEnabled = false
        }


        binding.button3.setOnClickListener {
            if (!isSaved && binding.editTextText.text.isNotEmpty()) {
                if (fileNameToSave == "New") {
                    showFilenameFragment()
                    state = 10
                } else {
                    saveToFile()
                    showFilelistIntent()
                    state = 8
                }
            } else {
                showFilelistIntent()
                state = 8
            }
            Log.d("myCheck", "MainActivity에서 '파일 열기' 버튼 클릭")
        }

        binding.editTextText.addTextChangedListener(object : TextWatcher {
            override fun beforeTextChanged(s: CharSequence?, start: Int, count: Int, after: Int) {}
            override fun onTextChanged(s: CharSequence?, start: Int, before: Int, count: Int) {
                isSaved = false
                binding.button2.isEnabled = true
                val tmp =
binding.textView2.text.toString().substring(binding.textView2.text.toString().length - 1)
                if (tmp != "*") {
                    binding.textView2.text = binding.textView2.text.toString() + "*"
                }
                if (state == 8 || state == 10) {
                    binding.textView2.text = fileNameToOpen
                    state = 1
                }
            }
            override fun afterTextChanged(s: Editable?) {}
        })

        supportFragmentManager.setFragmentResultListener("fileNameRequestKey", this) { requestKey,
bundle ->
            if (requestKey == "fileNameRequestKey") {
                val fileNameToSave = bundle.getString("fileNameToSave")
                if (!fileNameToSave.isNullOrEmpty()) {
                    this.fileNameToSave = fileNameToSave
                    binding.textView2.text = fileNameToSave
                    saveToFile()
                }
            }
        }
    }

    private fun showFilelistIntent() {
        filelistActivityResult.launch(Intent(this, FilelistActivity::class.java))}
```

```
    private fun saveToFile() {
        Log.d("myCheck", "파일 저장 시작 - 파일 이름은 ${fileNameToSave}")
        if (!isSaved) {
            val baseDir =
 Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_DOCUMENTS).toString()
            val file = File(baseDir, fileNameToSave)
            val fos = FileOutputStream(file)
            fos.write(binding.editTextText.text.toString().toByteArray())
            fos.close()
        }
        binding.button2.isEnabled = false
        isSaved = true
    }
```

```
    private fun openSelectedFile() {
        val baseDir =
Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_DOCUMENTS).toString()
        val file = baseDir + "/" + fileNameToOpen
        val reader = FileReader(file)
        val buffer = BufferedReader(reader)
        var temp: String?
        val readContent = StringBuilder()
        while (true) {
            temp = buffer.readLine()
            if (temp == null) break
            else readContent.append(temp).append("\n")
        }
        buffer.close()
        reader.close()
        Log.d("myCheck", "MainActivity에서 openSelectedFile 메서드 실행")
        Log.d("myCheck", "읽은 내용은 ${readContent}")
        binding.editTextText.setText(readContent.toString())
    }
```

```
    private fun showFilenameFragment() {
        val fragment = Filename.newInstance()
        val transaction: FragmentTransaction = supportFragmentManager.beginTransaction()
        transaction.replace(R.id.fragment_container, fragment)
        transaction.addToBackStack(null)
        transaction.commit()
    }
}
```

```
package com.android.memo_file

import android.app.Application

class MyApplication: Application() {

    companion object {
        lateinit var preferences: PreferenceUtil
    }

    override fun onCreate() {
        preferences = PreferenceUtil(applicationContext)
        super.onCreate()
    }
}
```

```
package com.android.memo_file

import android.content.Context
import android.content.SharedPreferences

class PreferenceUtil(context: Context) {

    private val preferences: SharedPreferences =
        context.getSharedPreferences("prefs_name", Context.MODE_PRIVATE)

    fun getString(key: String, prefs_value: String): String {
        return preferences.getString(key, prefs_value).toString()
    }

    fun setString(key: String, prefs_value: String) {
        preferences.edit().putString(key, prefs_value).apply()
    }
}
```