ГУАП

КАФЕДРА № 43

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Старший преподаватель |  |  |  | С.В. Щекин |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №6 |
| Организация навигации в многоэкранном приложении |
| по курсу: Программирование мобильных устройств |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4134к |  |  |  | Д.В. Самарин |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2024

Задание:

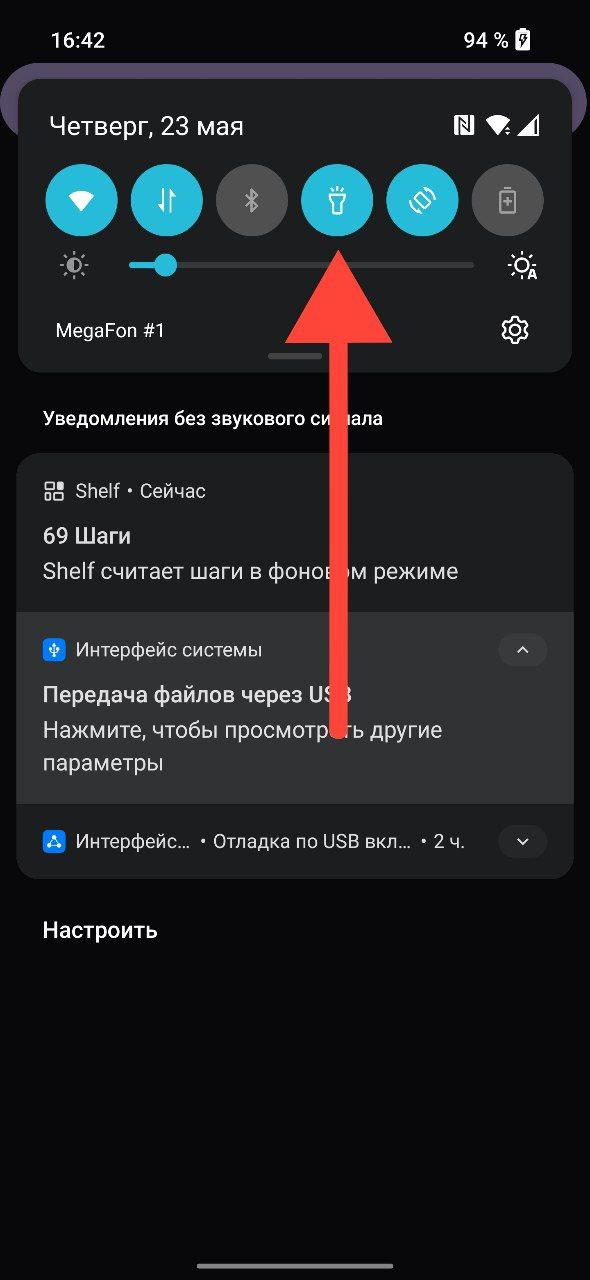
Дополнить приложение, разработанное в рамках одной из предыдущих лабораторных работ, добавив в него работу с аппаратным обеспечение мобильного устройства (камера, светодиод подсветки, датчики и т.п.)

Цель работы:

Получение навыков использования в разрабатываемых мобильных приложениия доступа к компонентам аппаратного обеспечения мобильных устройств.

Ход Работы

В ходе работы реализована работа с фонариком смартфона. В предыдущей работе уже были реализованы некоторые функции работы фонарика и камеры. Поэтому, доработки были минимальны.



## Листинг

|  |
| --- |
| MainActivity  package com.example.my\_inst    import android.content.Context import android.content.Intent import android.hardware.camera2.CameraAccessException import android.hardware.camera2.CameraCharacteristics import android.hardware.camera2.CameraManager import android.os.Bundle import android.view.View import android.widget.Button import androidx.activity.enableEdgeToEdge import androidx.appcompat.app.AppCompatActivity import androidx.core.view.ViewCompat import androidx.core.view.WindowInsetsCompat   class MainActivity : AppCompatActivity() {  private var isFlashlightOn = false   override fun onCreate(savedInstanceState: Bundle?) {  super.onCreate(savedInstanceState)  *enableEdgeToEdge*()  setContentView(R.layout.*activity\_main*)   ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.*main*)) **{** v, insets **->** val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())  v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)  insets  **}** val locationButton: Button = findViewById(R.id.*locationButton*)  locationButton.setOnClickListener **{** val intent = Intent(this, LocationActivity::class.*java*)  startActivity(intent)  **}** val flashlightButton: Button = findViewById(R.id.*flashlightButton*)  flashlightButton.setOnClickListener **{** toggleFlashlight()  **}** }   private fun toggleFlashlight() {  val cameraManager = getSystemService(Context.*CAMERA\_SERVICE*) as CameraManager  try {  val cameraId = cameraManager.*cameraIdList*[0]  val isFlashAvailable = cameraManager.getCameraCharacteristics(cameraId)  .get(CameraCharacteristics.*FLASH\_INFO\_AVAILABLE*)   if (isFlashAvailable == null || !isFlashAvailable) {  // Фонарик недоступен или отключен  return  }   isFlashlightOn = !isFlashlightOn  cameraManager.setTorchMode(cameraId, isFlashlightOn)  } catch (e: CameraAccessException) {  e.printStackTrace()  }  }   }  LocationActivity package com.example.my\_inst  import android.Manifest import android.annotation.SuppressLint import android.content.Context.*LOCATION\_SERVICE* import android.content.Intent import android.location.LocationManager import android.os.Bundle import android.view.View import android.widget.Button import android.widget.Toast  import androidx.activity.result.contract.ActivityResultContracts import androidx.appcompat.app.AppCompatActivity import androidx.core.content.ContextCompat.getSystemService import androidx.core.location.LocationManagerCompat.isLocationEnabled import com.example.my\_inst.databinding.ActivityLocationBinding  import com.example.my\_inst.databinding.ActivityMainBinding import com.google.android.gms.location.FusedLocationProviderClient import com.google.android.gms.location.LocationServices import com.google.android.gms.location.Priority import com.google.android.gms.location.LocationSettingsRequest import com.google.android.gms.location.LocationRequest  import com.google.android.gms.common.api.ResolvableApiException import com.google.android.gms.tasks.CancellationTokenSource     class LocationActivity : AppCompatActivity() {  private lateinit var binding: ActivityLocationBinding  private lateinit var fusedLocationClient: FusedLocationProviderClient   @SuppressLint("MissingPermission", "MissingInflatedId")  override fun onCreate(savedInstanceState: Bundle?)  {    super.onCreate(savedInstanceState)  setContentView(R.layout.*activity\_location*)   fun goToFlashlightActivity(view: View) {  val intent = Intent(this, MainActivity::class.*java*)  startActivity(intent)  }    binding = ActivityLocationBinding.inflate(*layoutInflater*)  val view = binding.*root* setContentView(view)   fusedLocationClient = LocationServices.getFusedLocationProviderClient(this)    val locationPermissionRequest= registerForActivityResult(  ActivityResultContracts.RequestMultiplePermissions()  ) **{** permissions **->** when {  permissions.getOrDefault(  Manifest.permission.*ACCESS\_FINE\_LOCATION*,  false  ) || permissions.getOrDefault(  Manifest.permission  .*ACCESS\_COARSE\_LOCATION*, false  ) -> {  Toast.makeText(  this, "loca access granted", Toast  .*LENGTH\_SHORT* ).show()   if (isLocationEnabled()) {  val result = fusedLocationClient.getCurrentLocation(  Priority.*PRIORITY\_BALANCED\_POWER\_ACCURACY*,  CancellationTokenSource().*token* )  result.addOnCompleteListener **{** val location =  "Latitude: " + **it**.*result*.*latitude* + "\n" + "Longitude: " +  **it**.*result*.*longitude* binding.textView.*text* = location  **}** } else {  Toast.makeText(  this, "Please turn on locatyion.",  Toast.*LENGTH\_SHORT* )  .show()  createLocationRequest()  }    }   else -> {  Toast.makeText(  this, "no loc access", Toast  .*LENGTH\_SHORT* ).show()   }  }  **}** binding.btnGetLocation.setOnClickListener **{** locationPermissionRequest.launch(  *arrayOf*(  Manifest.permission.*ACCESS\_FINE\_LOCATION*,  Manifest.permission.*ACCESS\_COARSE\_LOCATION* )  )  **}** }          private fun isLocationEnabled(): Boolean  {  val locationManager = getSystemService(*LOCATION\_SERVICE*) as LocationManager   try  {  return locationManager.isProviderEnabled(LocationManager.*GPS\_PROVIDER*)   } catch (e: Exception)  {  e.printStackTrace()  }  return false  }    private fun createLocationRequest()  {  val locationRequest = LocationRequest.Builder(  Priority.*PRIORITY\_HIGH\_ACCURACY*,  10000  ).setMinUpdateIntervalMillis(5000).build()   val builder = LocationSettingsRequest.Builder().addLocationRequest(locationRequest)   val client = LocationServices.getSettingsClient(this)   val task = client.checkLocationSettings(builder.build())   task.addOnCanceledListener **{   }** task.addOnFailureListener**{** e **->** if(e is ResolvableApiException){  try {  e.startResolutionForResult(  this,  100  )  }catch (sendEx: java.lang.Exception){  }  }  **}** } } |