

Включите вебку

#01



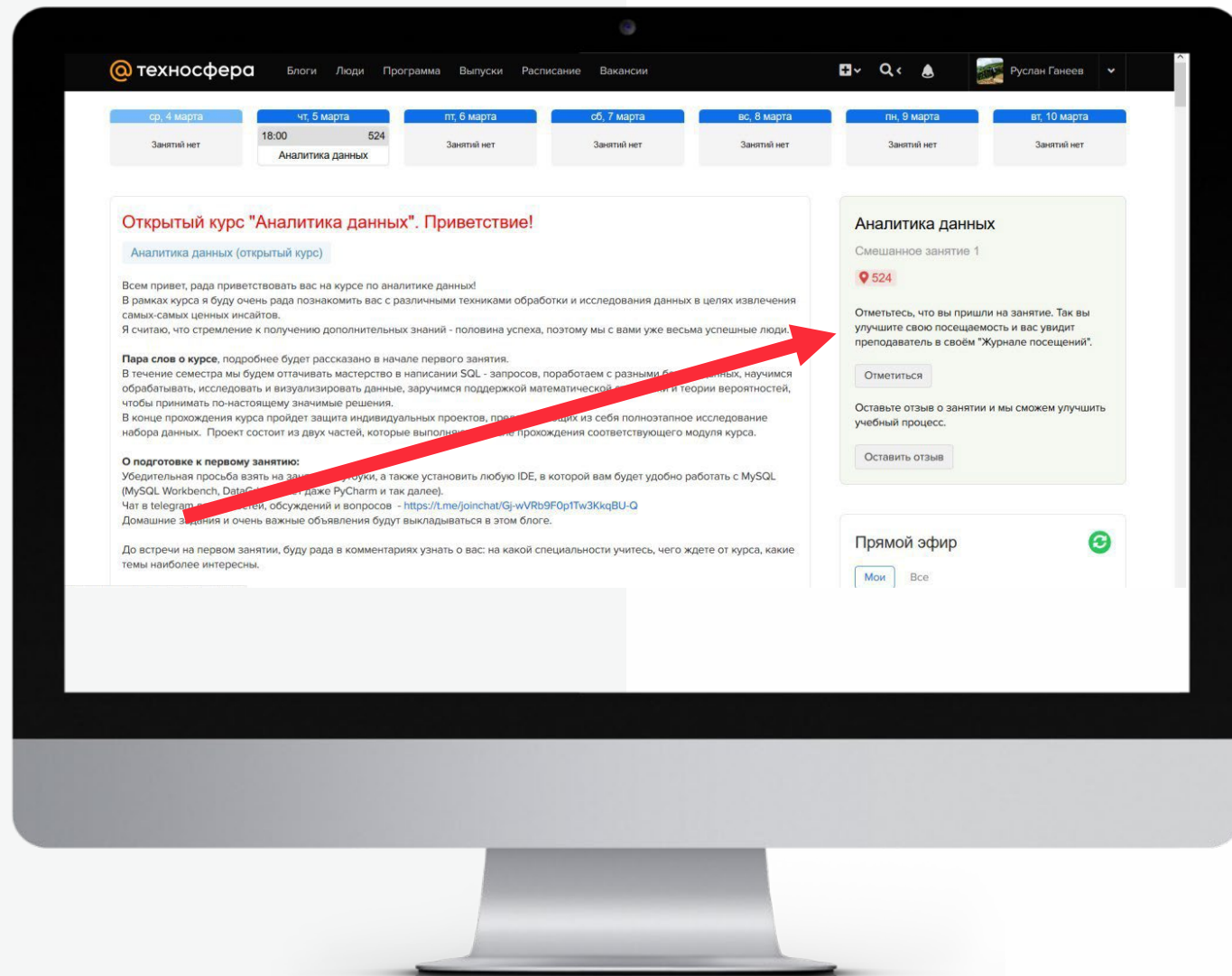


x @ mail.ru
group


МНОГОПОТОЧНОСТЬ

Куликов Никита



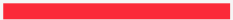


Напоминание
отметиться
на портале



ДЕЛОЙ ПРОЕКТЫ

1. Вступление



Содержание занятия

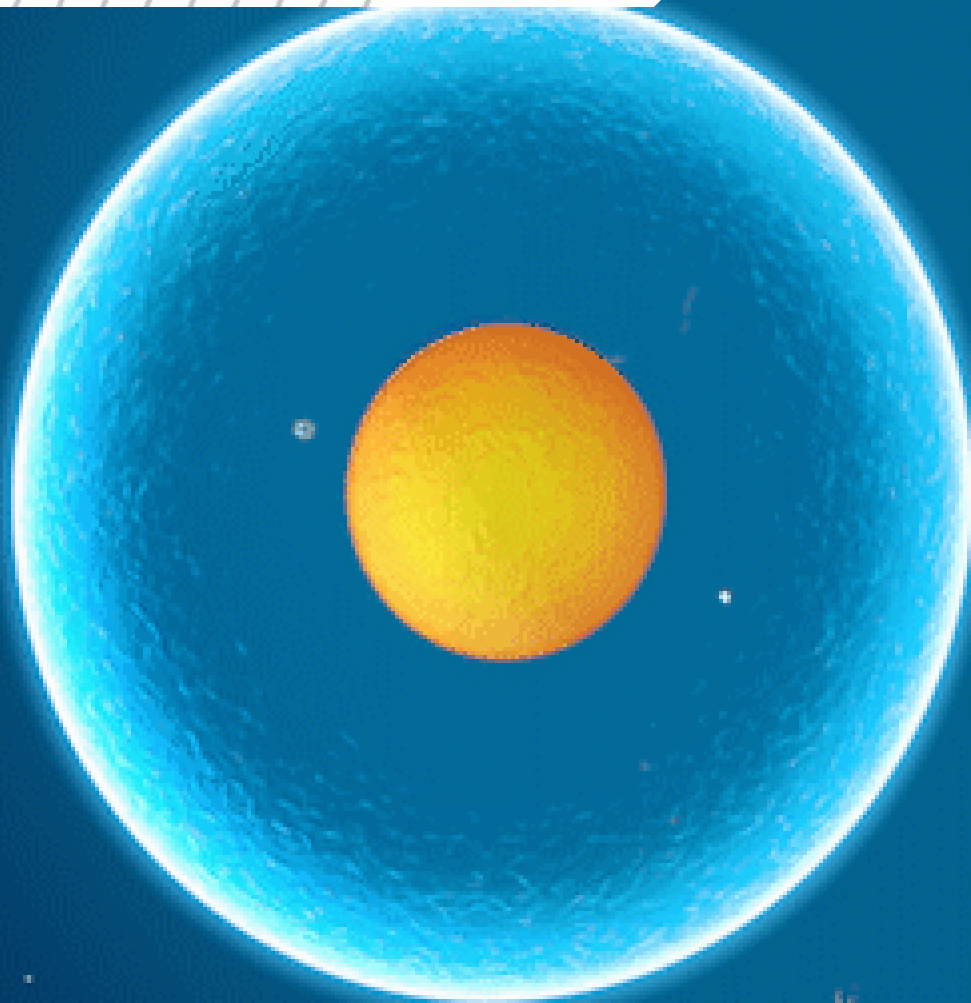
1. Вступление



Вступление

1. Как запускаются приложения
2. Тиковая система
3. 16 мс
4. Как бить себя по рукам
5. Почему все сложнее чем кажется
6. Выводы

Запуск приложений



Тиковая система

```
while (gameIsOpen()) {  
    draw(canvas)  
}
```




16 мс



#09

$$60\text{fps} = 1000\text{ms} / 60\text{f} = 16,6\text{ms}$$

frames per second

Should We want
MORE?

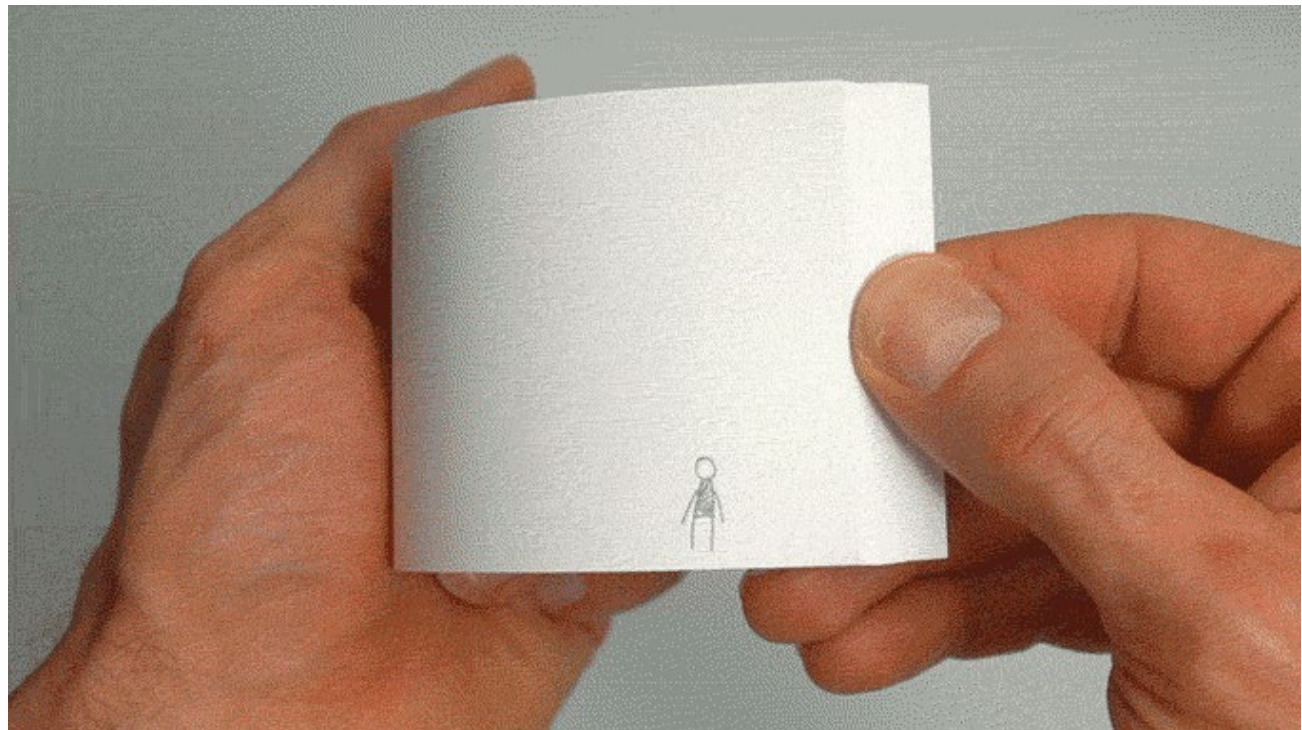
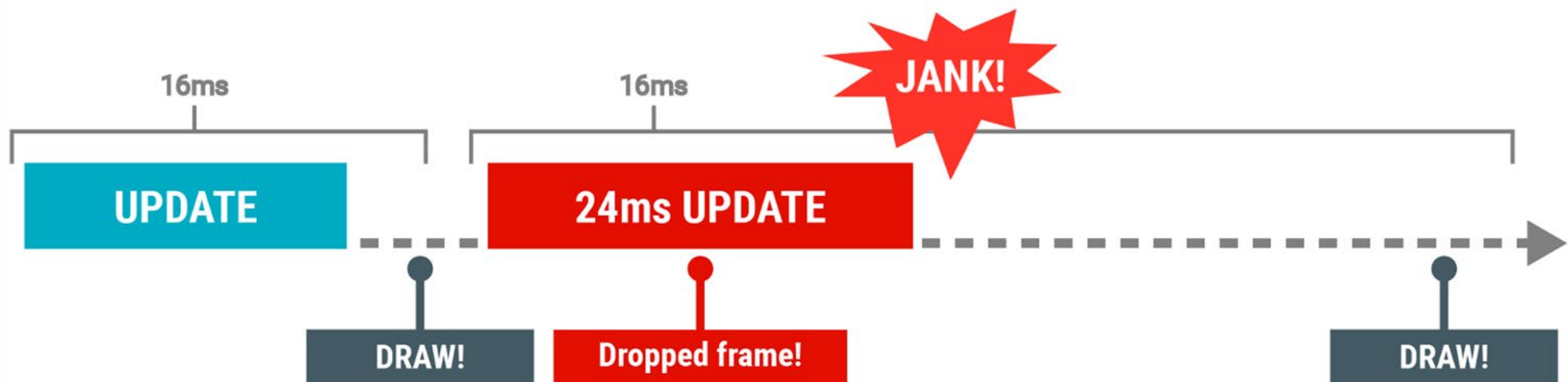
'PS

Looks good

'PS

Flipbook

'PS





Вот почему iPhone 12 не поддерживает 120 Гц

Я уже не пишу что у Самсунгов технология ЛТПО единичная на рынке и Айфоны даже банально 90 герц не способны сделать — уже не ...

1 месяц назад



Пять вещей, которые мы все хотели бы увидеть в Apple iPhone 12. Но не увидим

Сегодня вечером состоится презентация iPhone 12. ... Проще говоря, 90 Гц (и тем более 120 Гц) позволяют графике «поспевать» за ...

1 месяц назад



Чем iPhone 12 хуже многих флагманов на Android

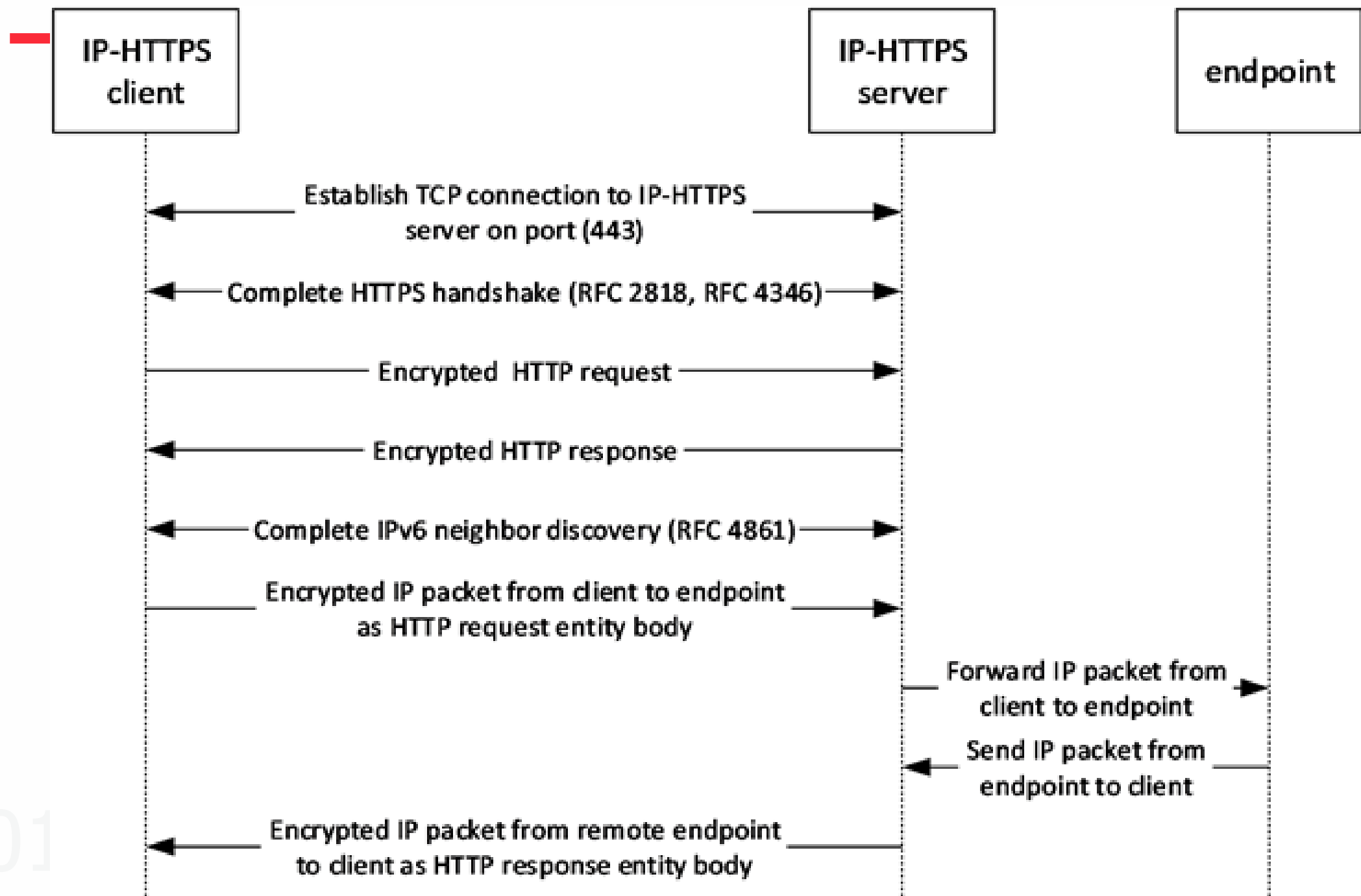
Вот как выглядит время зарядки разных аппаратов с выключенным режимом 90/120 Гц и включённым: Galaxy S20 5G без 120 Гц — 9 часов ...

3 недели назад



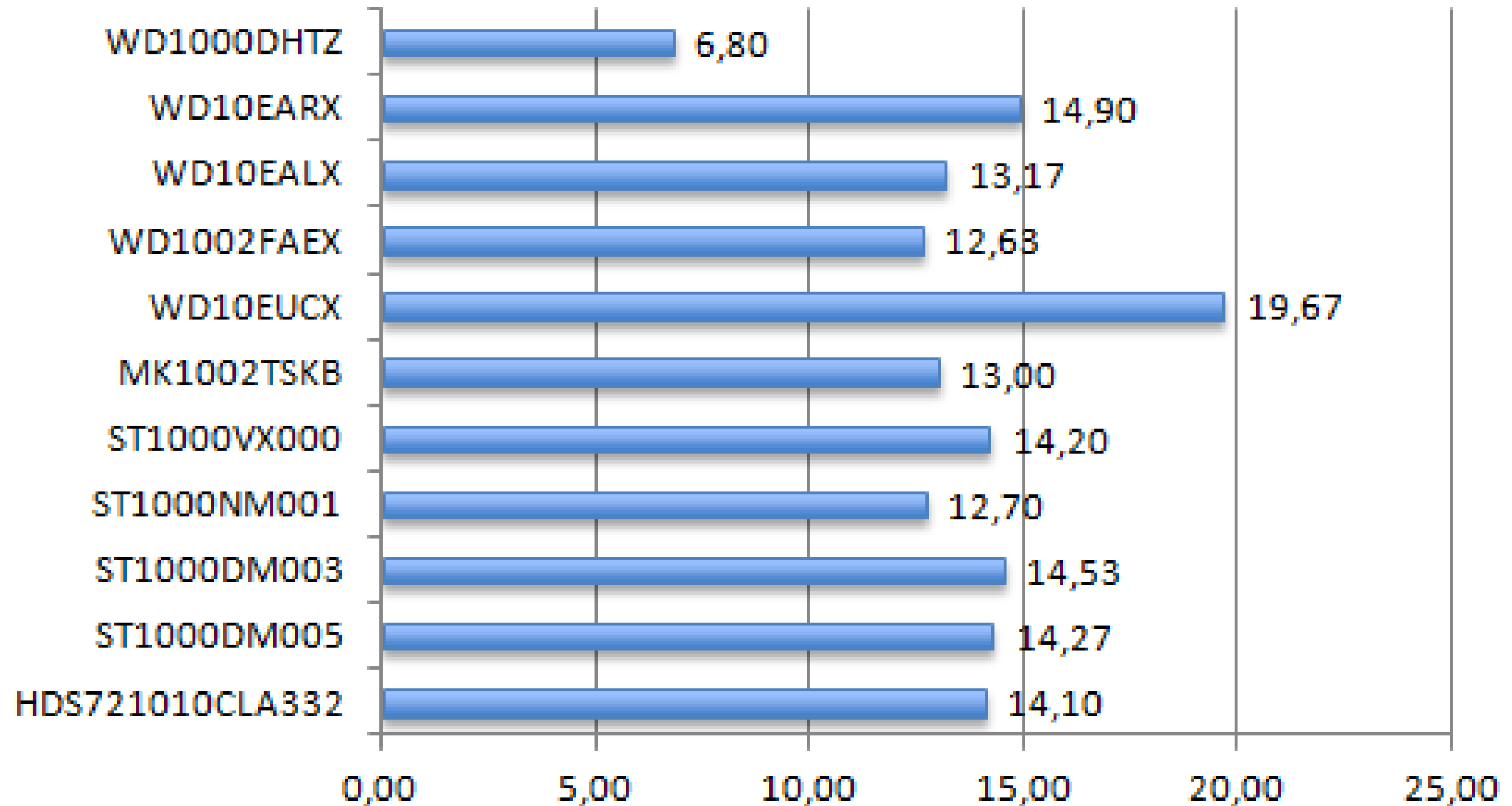
$$120\text{fps} = 1000\text{ms} / 120\text{f} = 8,3\text{ms}$$

frames per second



HDTach - Random Access Time

Werte in ms



```
class MainActivity : AppCompatActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
  
        val sharedPref =  
            getSharedPreferences("mySharedPrefs", Context.MODE_PRIVATE)  
        val firstEnter = sharedPref.getBoolean("firstEnter", true)  
  
        if (firstEnter) {  
            sharedPref.edit().putBoolean("firstEnter", false).commit()  
        }  
    }  
}
```

```
class MainActivity : AppCompatActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
  
        val sharedPref =  
            getSharedPreferences("mySharedPrefs", Context.MODE_PRIVATE)  
        val firstEnter = sharedPref.getBoolean("firstEnter", true)  
  
        if (firstEnter) {  
            sharedPref.edit().putBoolean("firstEnter", false).commit()  
        }  
    }  
}
```

```
class App : Application() {  
    override fun onCreate() {  
        super.onCreate()  
  
        if (BuildConfig.DEBUG) {  
            StrictMode.setThreadPolicy(  
                StrictMode.ThreadPolicy.Builder()  
                    .detectDiskReads()  
                    .detectDiskWrites()  
                    .detectNetwork() // or .detectAll() for all detectable problems  
                    .penaltyLog()  
                    .penaltyDeath()  
                    .build()  
            )  
            StrictMode.setVmPolicy(  
                VmPolicy.Builder()  
                    .detectLeakedSqlLiteObjects()  
                    .detectLeakedClosableObjects()  
                    .penaltyLog()  
                    .penaltyDeath()  
                    .build()  
            )  
        }  
    }  
}
```

#

```
class App : Application() {  
    override fun onCreate() {  
        super.onCreate()  
  
        if (BuildConfig.DEBUG) {  
            StrictMode.setThreadPolicy(  
                StrictMode.ThreadPolicy.Builder()  
                    .detectDiskReads()  
                    .detectDiskWrites()  
                    .detectNetwork() // or .detectAll() for all detectable problems  
                    .penaltyLog()  
                    .penaltyDeath()  
                    .build()  
            )  
            StrictMode.setVmPolicy(  
                VmPolicy.Builder()  
                    .detectLeakedSqlLiteObjects()  
                    .detectLeakedClosableObjects()  
                    .penaltyLog()  
                    .penaltyDeath()  
                    .build()  
            )  
        }  
    }  
}
```

#

----- beginning of crash

2020-11-10 16:08:37.439 12120-12120/com.example.multithread E/AndroidRuntime: FATAL EXCEPTION: main

Process: com.example.multithread, PID: 12120

java.lang.RuntimeException: StrictMode ThreadPolicy violation

at android.os.StrictMode\$AndroidBlockGuardPolicy.onThreadPolicyViolation([StrictMode.java:1813](#))

at android.os.StrictMode\$AndroidBlockGuardPolicy.lambda\$handleViolationWithTimingAttempt\$0\$StrictMode\$AndroidBlockGuardPolicy\$1.run(StrictMode.java:1813)

at android.os.-\$\$Lambda\$StrictMode\$AndroidBlockGuardPolicy\$9nBulCQKaMajrWr41SB7f7YRT1I.run(Unknown Source:6)

at android.os.Handler.handleCallback([Handler.java:938](#))

at android.os.Handler.dispatchMessage([Handler.java:99](#))

at android.os.Looper.loop([Looper.java:223](#))

at android.app.ActivityThread.main(ActivityThread.java:7656) **<1 internal call>**

at com.android.internal.os.RuntimeInit\$MethodAndArgsCaller.run(RuntimeInit.java:592)


at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:947)

Caused by: android.os.strictmode.DiskReadViolation

at android.os.StrictMode\$AndroidBlockGuardPolicy.onReadFromDisk([StrictMode.java:1596](#))

at libcore.io.BlockGuardOs.access(BlockGuardOs.java:71)

at libcore.io.BlockGuardOs.access(BlockGuardOs.java:71)



```
class MainActivity : AppCompatActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
  
        sharedPreferences.edit().putBoolean("firstEnter", false).apply()  
    }  
}
```




UI Thread

The diagram illustrates a flow from the UI Thread to SharedPrefs. A teal box labeled 'UI Thread' has a downward-pointing arrow leading to a teal box labeled 'SharedPrefs'. The 'SharedPrefs' box is partially filled with red on its right side. The background features a light gray area with a diagonal line pattern on the right.

SharedPrefs

#024



Многопоточность в Android – ~~это ж...~~ важна

#025





Как детектить?

1. StrictMode
2. ANR
3. Developer options



StrictMode



#027

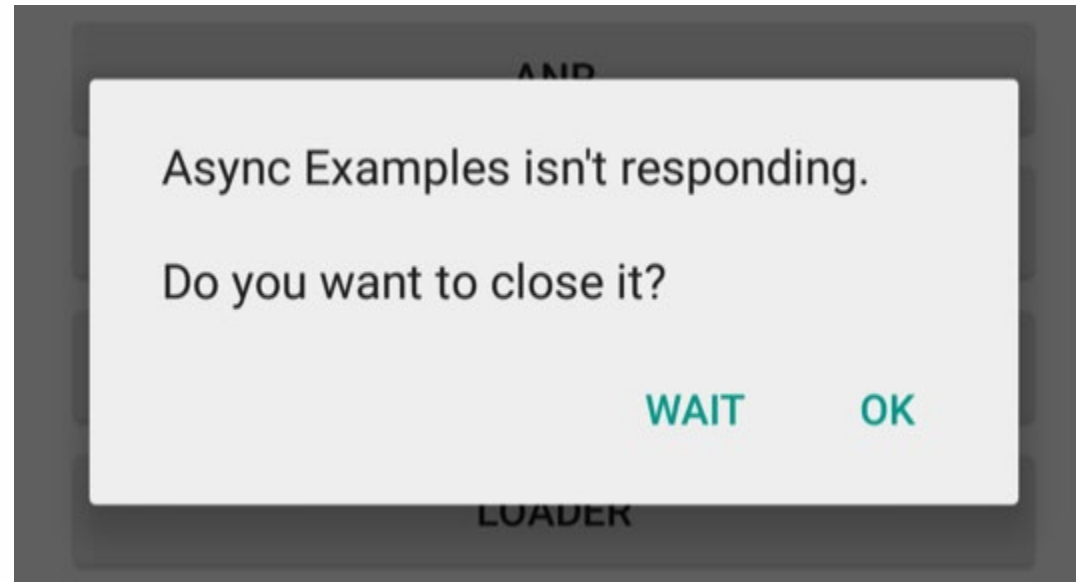


ANR



#028

Application Not Responding

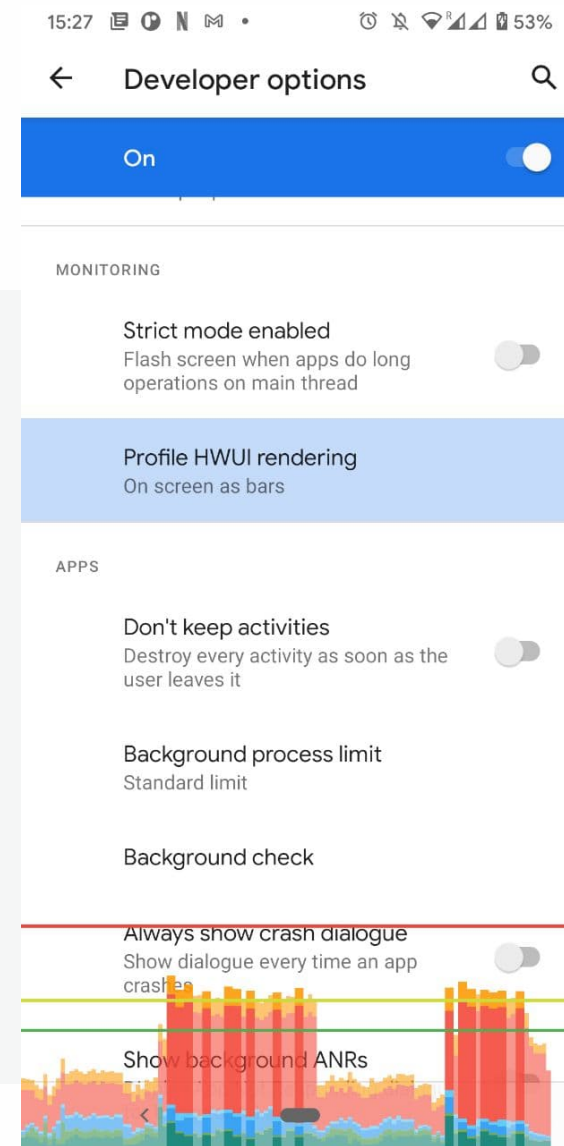
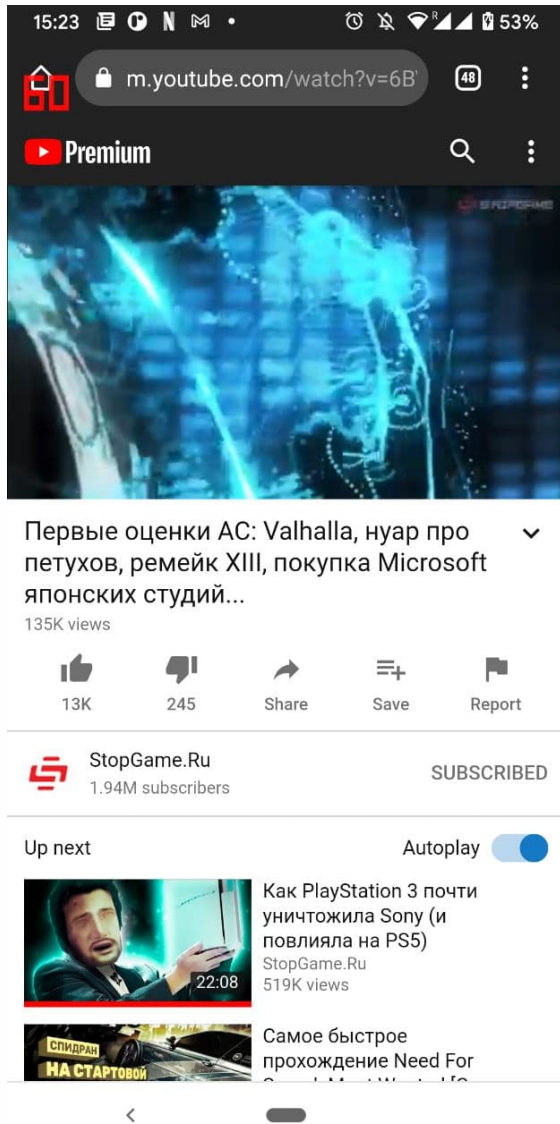





Developers options



#030



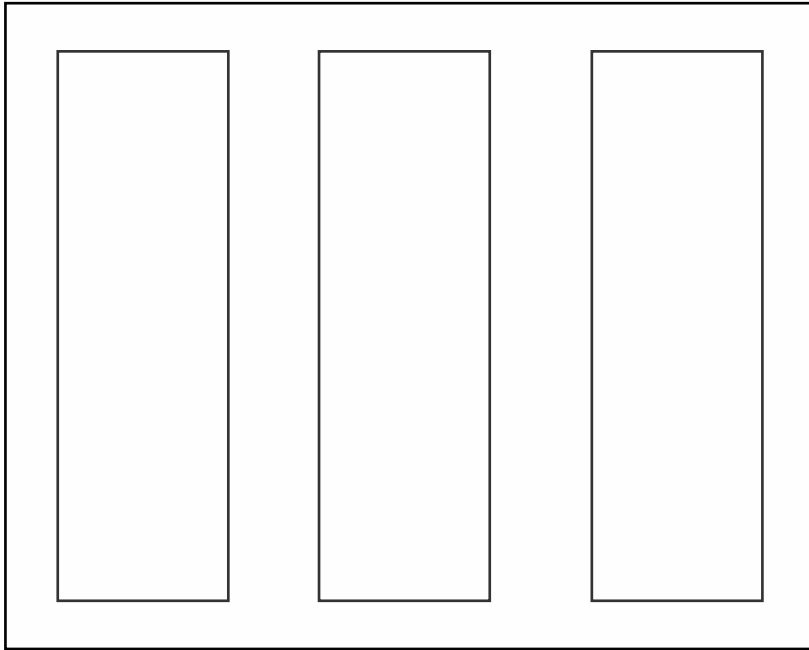
#031



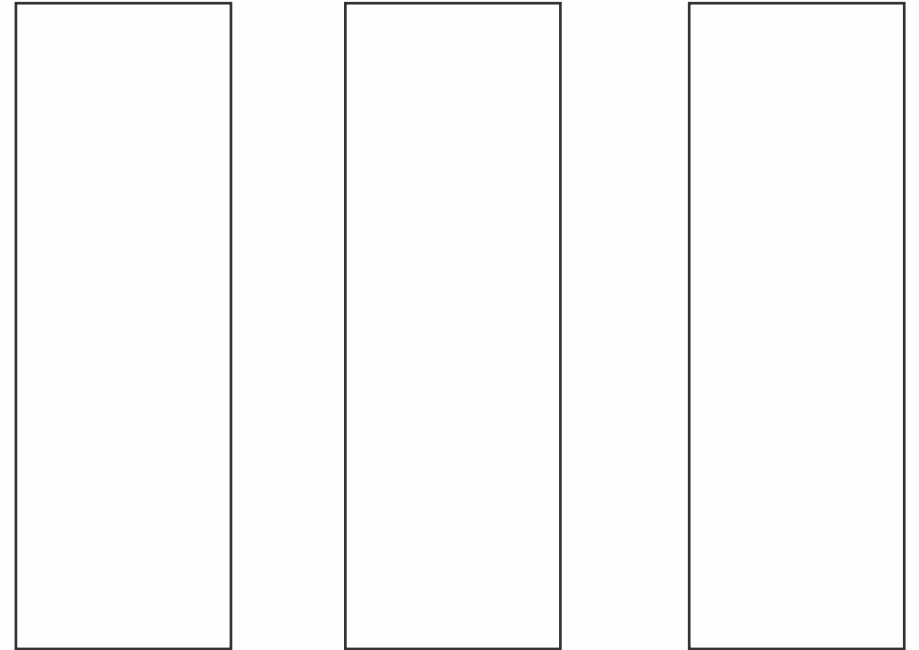
Выносим в параллель

1. Как запускаются приложения
2. Тиковая система
3. 16 мс
4. Как бить себя по рукам
5. Почему все сложнее чем кажется
6. Выводы

Потоки



Процессы



Процессы: AIDL это не ругательство



Йонатан Левин

GETT

IPC: AIDL — это
не ругательство



<https://www.youtube.com/watch?v=sBlIMiaA8to>

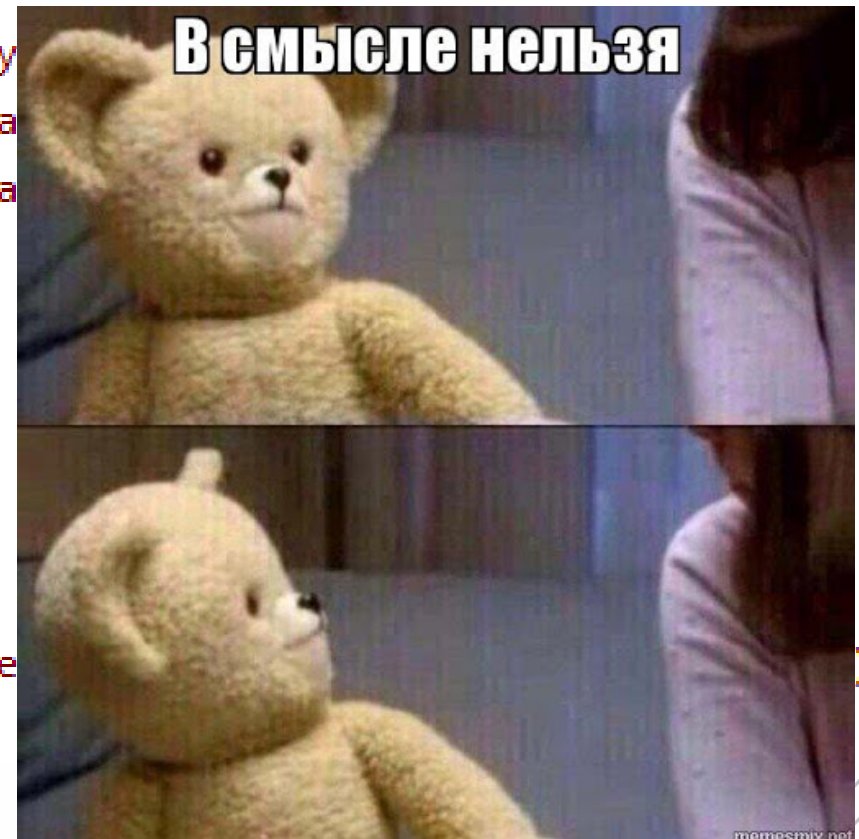
#034

Окей, ща наварганим

```
override fun onCreate(savedInstanceState: Bundle?) {  
    super.onCreate(savedInstanceState)  
    setContentView(R.layout.activity_main)  
  
    val textView = findViewById<TextView>(R.id.text_from_ui)  
  
    Thread(Runnable {  
        val text = downloadText()  
        textView.setText(text)  
    }).start()  
}
```

В смысле нельзя?

```
2020-11-10 17:22:55.146 24453-24514/com.example.multithread E/AndroidRuntime: FATAL EXCEPTION: Thread-2
Process: com.example.multithread, PID: 24453
android.view.ViewRootImpl$CalledFromWrongThreadException: Only
    at android.view.ViewRootImpl.checkThread(ViewRootImpl.java
    at android.view.ViewRootImpl.requestLayout(ViewRootImpl.ja
    at android.view.View.requestLayout(View.java:25390)
    at android.view.View.requestLayout(View.java:25390)
    at android.view.View.requestLayout(View.java:25390)
    at android.view.View.requestLayout(View.java:25390)
    at android.view.View.requestLayout(View.java:25390)
    at android.view.View.requestLayout(View.java:25390)
    at android.view.View.requestLayout(View.java:25390)
    at androidx.constraintlayout.widget.ConstraintLayout.reque
    at android.view.View.requestLayout(View.java:25390)
```



Хорошо, а так?

```
runOnUiThread {  
    textView.setText(text)  
}
```



<https://github.com/LionZXY/TestForMobilization/>

Школа мобильной разработки: ответ на заявку

job@yandex-team.ru <job@yandex-team.ru>
Кому: nikita@kulikof.ru

23 мая 2016 г., 14:18

Здравствуйте, Никита.

Мы рассмотрели все заявки, еще раз внимательно оценили итоги проверки вашего тестового задания и, к сожалению, вынуждены отказать: ваша работа оказалась недостаточно сильной.

Спасибо за проявленный интерес и время, потраченное на выполнение заданий.

Не останавливайтесь в развитии в выбранном направлении - мы считаем, у вас есть потенциал, и будем ждать ваших откликов в будущем.

Желаем вам удачи!

Команда Школы мобильной разработки

#038

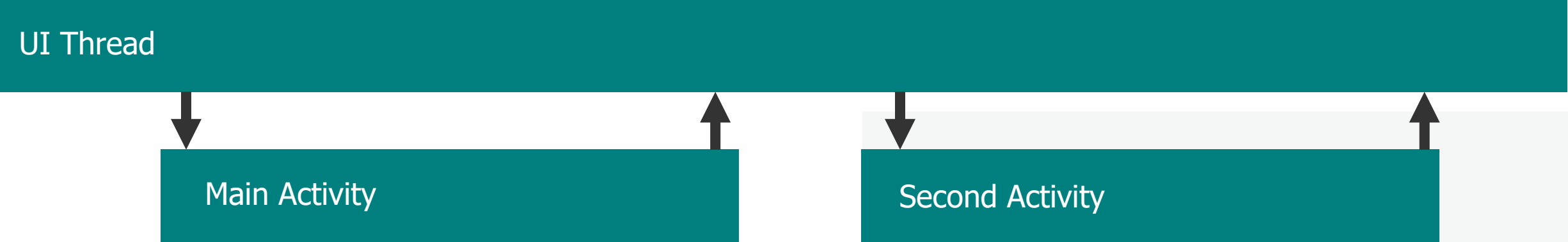


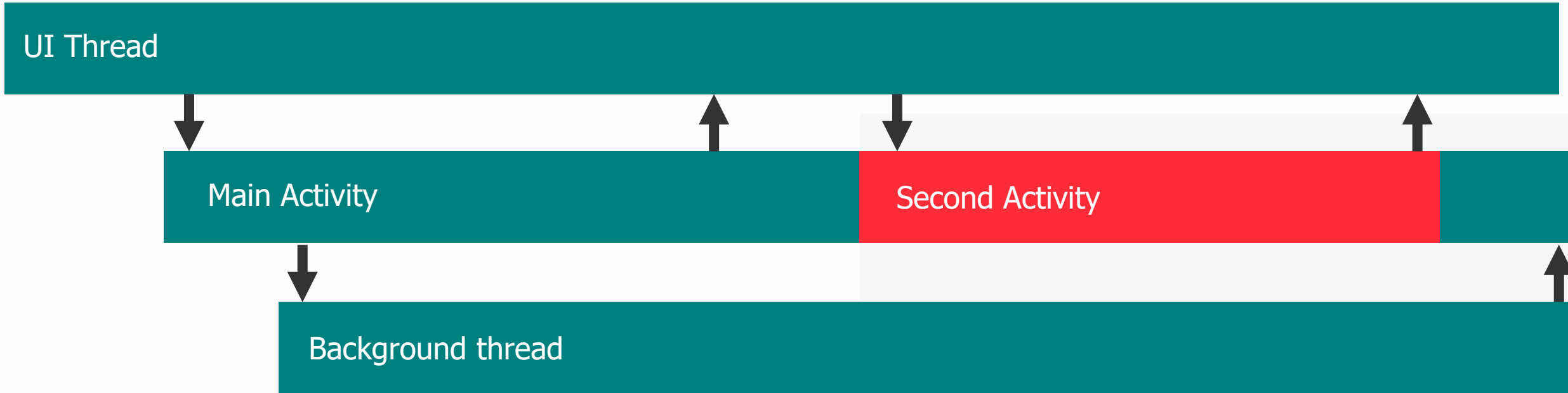



**Как люди
представляют процесс
духовного роста**



**Как это выглядит
на самом деле**







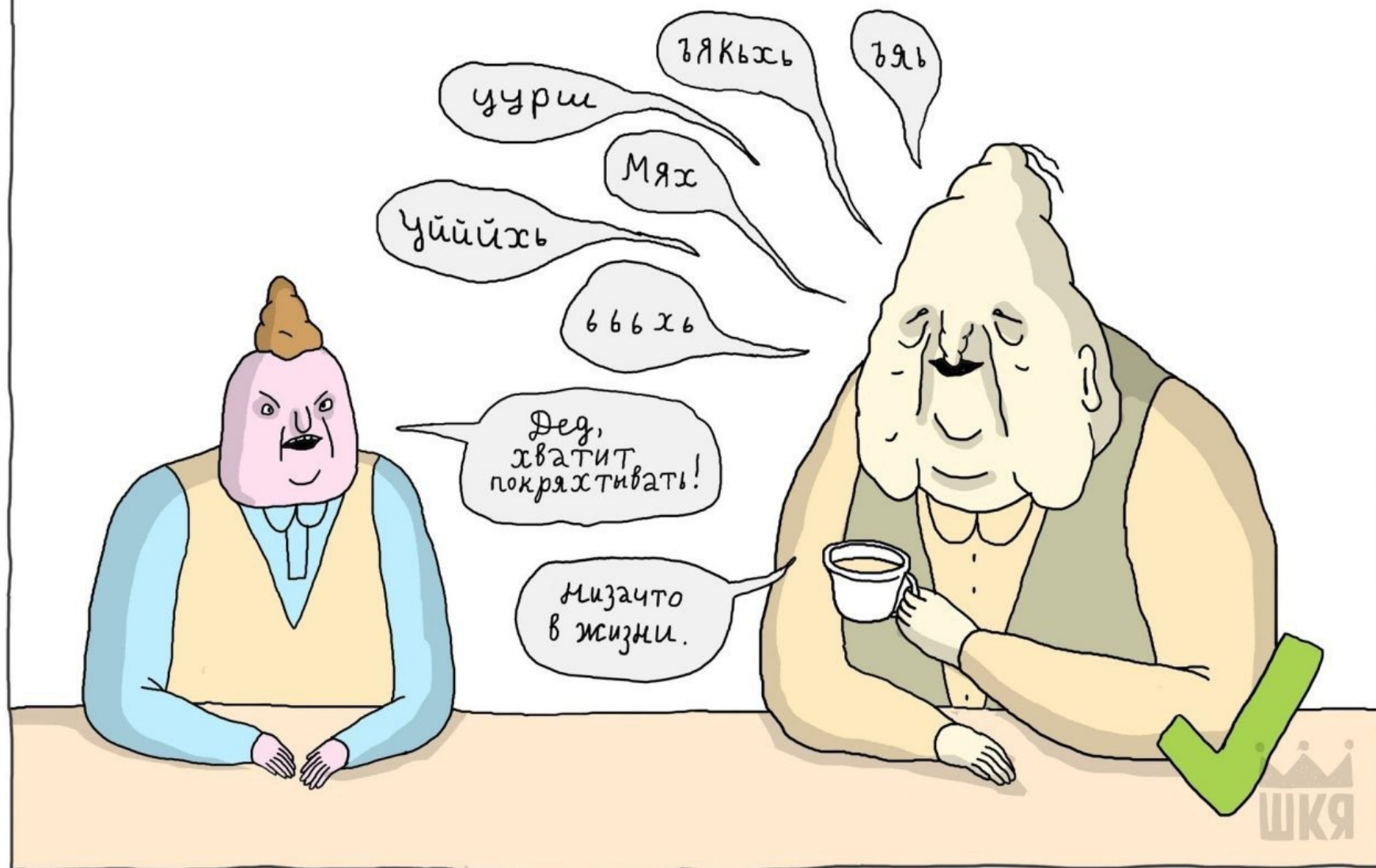
Делаем
нормально

#042



Как делали деды

3. Дед всегда должен покряхтивать.



Отдельный класс

```
class DownloadTask : Runnable {  
    override fun run() {  
        val text = downloadText()  
    }  
}
```

Добавим слушатель

```
class DownloadTask : Runnable {  
    private var listener: IDownloadListener? = null  
  
    override fun run() {  
        val text = downloadText()  
        listener?.onDownloadObject(text)  
    }  
}
```

И подписку/отписку

```
fun subscribe(listener: IDownloadListener) {  
    this.listener = listener  
}  
  
fun unsubscribe() {  
    this.listener = null  
}
```

Итого


```
class DownloadTask : Runnable {  
    private var listener: IDownloadListener? = null  
  
    override fun run() {  
        val text = downloadText()  
        listener?.onDownloadObject(text)  
    }  
  
    fun subscribe(listener: IDownloadListener) {  
        this.listener = listener  
    }  
  
    fun unsubscribe() {  
        this.listener = null  
    }  
}
```

Подписываемся/отписываемся

```
class MainActivity : AppCompatActivity(), IDownloadListener {  
    val downloadTask = DownloadTask()  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        downloadTask.subscribe(this)  
        Thread(downloadTask).start()  
    }  
  
    override fun onDestroy() {  
        downloadTask.unsubscribe()  
    }  
}
```

Наследуемся от интерфейса

```
override fun onDownloadObject(text: String) {  
    runOnUiThread {  
        textView.setText(text)  
    }  
}
```



Улучшаем



#050

Делаем WeakReference

```
class DownloadTask : Runnable {  
    private var listener = WeakReference<IDownloadListener>(null)  
}
```

Добавляем handler

```
val handler = Handler(Looper.getMainLooper())  
  
override fun run() {  
    val text = downloadText()  
    handler.post {  
        listener.get()?.onDownloadObject(text)  
    }  
}
```

Что за магия?

#053

Еще
улучшаем

#054

Executor


#055

Service

#056

Не используйте AsyncTask

#057



А можно
попроще?

1. RxJava
2. Корутины



Можно



#059

Что из этого покемоны?

A word cloud of Kotlin-related libraries and frameworks, presented with a Pokemon theme. The words are arranged in a scattered, non-uniform layout. The words 'electabuzz' and 'dodrio' are the largest and are highlighted with a light gray background. Other words include 'rnetwork', 'fuel', 'rxjava', 'kine', 'sandslash', 'merlin', 'coroutines', 'okhttp', 'charizard', 'lon', 'nidoking', and 'rnetwork'. The background is white with a subtle gray grid pattern.

kine rxjava sandslash merlin
coroutines
rnetwork
fuel
okhttp
nidoking lon
electabuzz
charizard
dodrio

#060

RxJava

#061

Корутины

#062

**СПАСИБО
ЗА ВНИМАНИЕ**

