

## Практика 6: Приложение секундомер

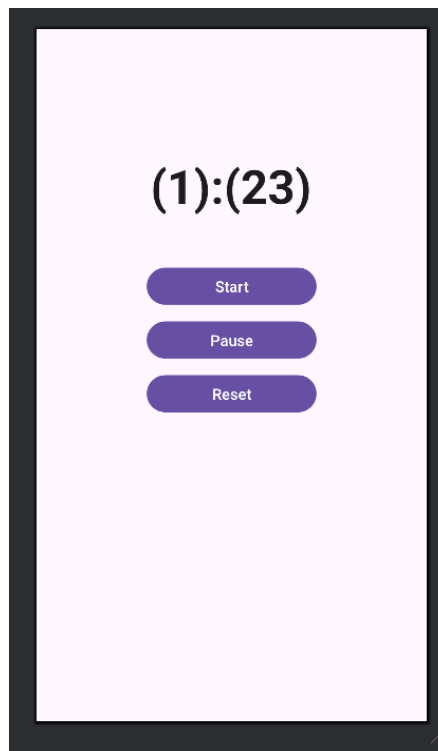
Создаем новый проект

SK Sekundomer\_Kostya ▾

Добавляем строковые ресурсы

```
<resources>
  <string name="app_name">Sekundomer_Kostya</string>
  <string name="start">Start</string>
  <string name="pause">Pause</string>
  <string name="reset">Reset</string>
</resources>
```

Делаем макет без подсказок



Объявление переменных в классе MainActivity

```
class MainActivity : AppCompatActivity() {
    lateinit var binding: ActivityMainBinding
    lateinit var chronometer: Chronometer
    var running = false
    var offset = 0L
}
```

Инициализация в функции onCreate

```
//Инициализация
chronometer = binding.textTime
val btnStart = binding.start
val btnPause = binding.pause
val btnReset = binding.reset
```

## Логика поведения кнопок

```
//Кнопка start запускает секундомер
btnStart.setOnClickListener{
    if(!running){
        setBaseTime()
        chronometer.start()
        running = true
    }
}

//Кнопка pause останавливает секундомер
btnPause.setOnClickListener{
    if(running){
        saveOffset()
        chronometer.stop()
        running = false
    }
}

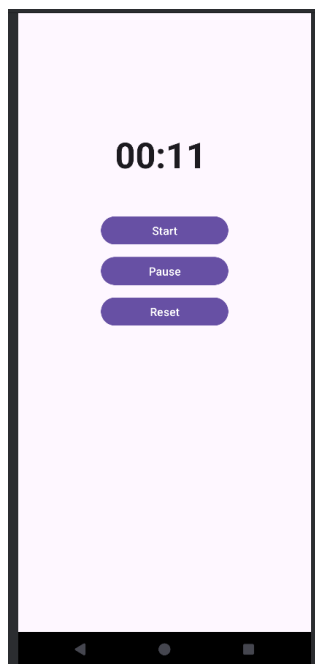
//Кнопка reset обнуляет время
btnReset.setOnClickListener{
    chronometer.stop()
    offset = 0
    chronometer.base = SystemClock.elapsedRealtime()
    running = false
}
```

## Функции

```
private fun saveOffset(){
    offset = SystemClock.elapsedRealtime() - chronometer.base
}

private fun setBaseTime(){
    chronometer.base = SystemClock.elapsedRealtime() - offset
}
```

## Проверяем работоспособность



Код для восстановления

```
//Восстановление предыдущего состояния
if(savedInstanceState != null){
    offset = savedInstanceState.getLong(OFFSET_K)
    running = savedInstanceState.getBoolean(RUNNING_K)
    if(running){
        chronometer.base = savedInstanceState.getLong(BASE_K)
        chronometer.start()
    } else setBaseTime()
}
```

Переопределим функцию сохранения состояния

```
override fun onSaveInstanceState(outState: Bundle) {
    outState.putLong(OFFSET_K, offset)
    outState.putBoolean(RUNNING_K, running)
    outState.putLong(BASE_K, chronometer.base)
    super.onSaveInstanceState(outState)
}
```

Теперь повороты экрана не страшны

