import kotlinx.coroutines.\*  
  
fun main() = runBlocking {  
// Запуск в фоновом потоке  
val backgroundThread = Thread {  
Thread.sleep(1000)  
println("World")  
}  
backgroundThread.start()  
  
// Печать в основном потоке с блокировкой  
Thread.sleep(2000)  
println("Hello,")  
  
// Функции с задержкой  
fun function1(): Int {  
Thread.sleep(1000) // Задержка 1 секунда  
return 10  
}  
  
fun function2(): Int {  
Thread.sleep(1000) // Задержка 1 секунда  
return 20  
}  
  
// Последовательный вызов  
val startTimeSequential = System.currentTimeMillis()  
val resultSequential = function1() + function2()  
println("Последовательный результат: $resultSequential. Время работы: ${System.currentTimeMillis() - startTimeSequential} ms")  
  
// Асинхронный вызов  
val startTimeAsync = System.currentTimeMillis()  
val resultAsync = async { function1() } + async { function2() }  
println("Асинхронный результат: ${resultAsync.await()}. Время работы: ${System.currentTimeMillis() - startTimeAsync} ms")  
  
// Дополнительный текст  
launch {  
repeat(3) { i ->  
println("I'm sleeping $i ...")  
delay(1000)  
}  
println("main: I'm tired of waiting! I'm running finally")  
}.join()  
println("main: Now I can quit.")  
}