**Задача 12**

Цели практической работы

* Закрепить на практике работу с корутинами.
* Создать приостанавливаемые функции.
* Научиться запускать и прерывать исполнение корутин.

Что нужно сделать

Для работы с корутинами мы будем использовать уже знакомую вам задачу о числах Фибоначчи с использованием циклов.

Необходимо доработать существующий код или написать программу заново с использованием корутин. При возникновении сложностей обратитесь за помощью к куратору или к блоку «Советы и рекомендации».

    1. Создайте объект-синглтон Fibonacci.

    2. В этом объекте создайте suspend функцию take. Функция принимает на вход число — номер в последовательности Фибоначчи — и возвращает рассчитанное число типа BigInteger. Для создания числа BigInteger используйте функцию-расширение [toBigInteger](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/to-big-integer.html" \t "_blank).

    3. Запустите две или более корутин в функции main. Используйте для этого функцию runBlocking и билдер launch. Внутри каждой корутины вызовите функцию take и выведите результат в консоль.

    4. Рассчитайте несколько чисел Фибоначчи в порядке возрастания, затем в порядке убывания. Запустите программу несколько раз и проанализируйте порядок работы корутин.

*Например:*

*По возрастанию:*

launch { take (5) }   
launch { take (10) }  
launch { take (15) }

    5. Сделайте suspend функцию take отменяемой. Используйте для этого функцию [yield](https://kotlin.github.io/kotlinx.coroutines/kotlinx-coroutines-core/kotlinx.coroutines/yield.html" \t "_blank). Или функцию [currentCoroutineContext](https://kotlin.github.io/kotlinx.coroutines/kotlinx-coroutines-core/kotlinx.coroutines/current-coroutine-context.html" \t "_blank) и его расширение [isActive](https://kotlin.github.io/kotlinx.coroutines/kotlinx-coroutines-core/kotlinx.coroutines/is-active.html" \t "_blank).

*Например:*

currentCoroutineContext().isActive

При необходимости выполнение функции должно прерываться. Точки прерывания определите самостоятельно, исходя из реализации вашей функции.

    6. Запустите программу ещё несколько раз и вновь проанализируйте порядок работы корутин.

    7. Отмените выполнение функции take при превышении какого-то времени. Используйте для этого функцию [withTimeout](https://kotlin.github.io/kotlinx.coroutines/kotlinx-coroutines-core/kotlinx.coroutines/with-timeout.html" \t "_blank).

Например:

withTimeout(3000) {  
    *// some code*  
*}*

запускает блок кода на выполнение и прерывает его выполнение при превышении 3-х секунд.

Подробнее про прерывание по таймауту можно прочитать [здесь](https://kotlinlang.org/docs/cancellation-and-timeouts.html#timeout).

Числа в цикле рассчитываются быстро, поэтому, чтобы отмена по таймауту была заметна, необходимо рассчитывать числа из последовательности Фибоначчи с довольно большим порядковым номером. Использование цикла вместо рекурсии и BigInteger вместо Long избавит от переполнения типа данных и переполнения стека.

    8. При отмене по таймауту выбрасывается исключение TimeoutCancellationException. Обработайте это исключение и выведите в консоль сообщение о превышении времени вычисления.

    9. Запустите ещё одну корутину, которая будет индикатором вычислений и выведет точки в консоль. Эта корутина должна работать до тех пор, пока идут вычисления всех ваших чисел Фибоначчи.

**\* По желанию:** Добейтесь, чтобы вывод рассчитанных значений происходил после того, как будут рассчитаны все числа. Тем самым числа будут рассчитываться параллельно, но индикатор прогресса (точки) не будет прерываться сообщениями о рассчитанном числе.

Можно использовать для этого [Job](https://kotlin.github.io/kotlinx.coroutines/kotlinx-coroutines-core/kotlinx.coroutines/-job/" \t "_blank) и функцию [join](https://kotlin.github.io/kotlinx.coroutines/kotlinx-coroutines-core/kotlinx.coroutines/-job/join.html" \t "_blank).

Советы и рекомендации

Воспользуйтесь документацией:

* [Общая информация по корутинам](https://kotlinlang.org/docs/coroutines-basics.html).
* Тип [BigInteger](https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/java.math.-big-integer/" \t "_blank).
* Функция [yield](https://kotlin.github.io/kotlinx.coroutines/kotlinx-coroutines-core/kotlinx.coroutines/yield.html" \t "_blank).
* Функция [currentCoroutineContext](https://kotlin.github.io/kotlinx.coroutines/kotlinx-coroutines-core/kotlinx.coroutines/current-coroutine-context.html" \t "_blank).
* [Отмена корутин](https://kotlinlang.org/docs/cancellation-and-timeouts.html).
* [Job](https://kotlin.github.io/kotlinx.coroutines/kotlinx-coroutines-core/kotlinx.coroutines/-job/).