|  |
| --- |
| https://lh6.googleusercontent.com/QcftzNtI05T0Y6fjdSh1Rr2rt8oqZ1IvnLvbn1jLJ7CCyteVir3k-xBLv4SL1wAgWJsRhmmJSR0UW-RP63_GQenE4vVWv05BRoZTsmIcBccVTnfxwmsnNMvjg599x9SqZd8E3dkd |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение высшего образования  **«МИРЭА - Российский технологический университет»**  **РТУ МИРЭА** |

Институт информационных технологий (ИТ)

Кафедра инструментального и прикладного программного обеспечения (ИиППО)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ОТЧЕТ ПО ПРАКТИЧЕСКИМ РАБОТАМ** | | | | | |
| **по дисциплине «Разработка клиент-серверных приложений»** | | | | | |
|  | | | | | |
|  | | | | | |
| Выполнил студент группы ИКБО-20-19 | | | | Московка А.А. | |
|  | | | |  | |
| Принял  *Ассистент* | | Зарипов Е.А. | | |
| Практические работы выполнены | «\_\_\_»\_\_\_\_\_\_\_2022 г. | | (подпись студента) | | |
| «Зачтено» | «\_\_\_»\_\_\_\_\_\_\_2022 г. | | (подпись руководителя) | | |
|  |  | |  | | |

Москва 2022

**Практическая работа №1: Примитивы синхронизации и их использование при создании клиент серверных приложений**

**Ход работы:** для решения первой практической работы были написаны класс Main (Рисунок 1), три класса для функций (Рисунки 2-5), классы для реализации с помощью многопоточности (Рисунок 6) и с использованием метода ForkJoin (Рисунок 7), класс с используемыми инструментами Tools (Рисунок 8).

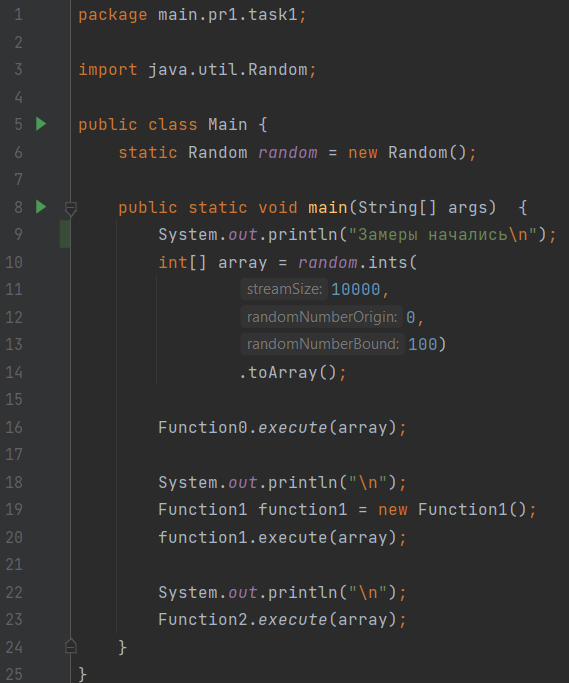


Рисунок 1 – Скриншот кода класса Main

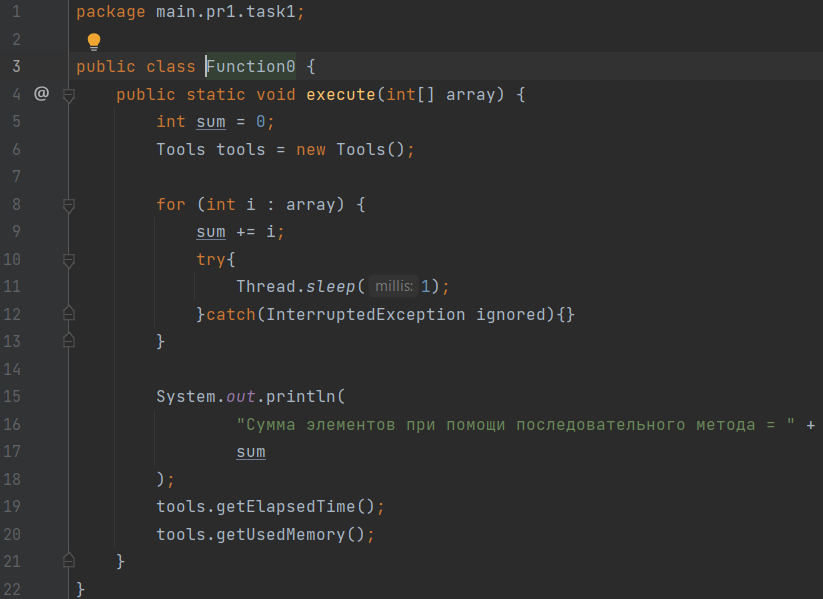


Рисунок 2 – Скриншот кода класса императивного решения задачи

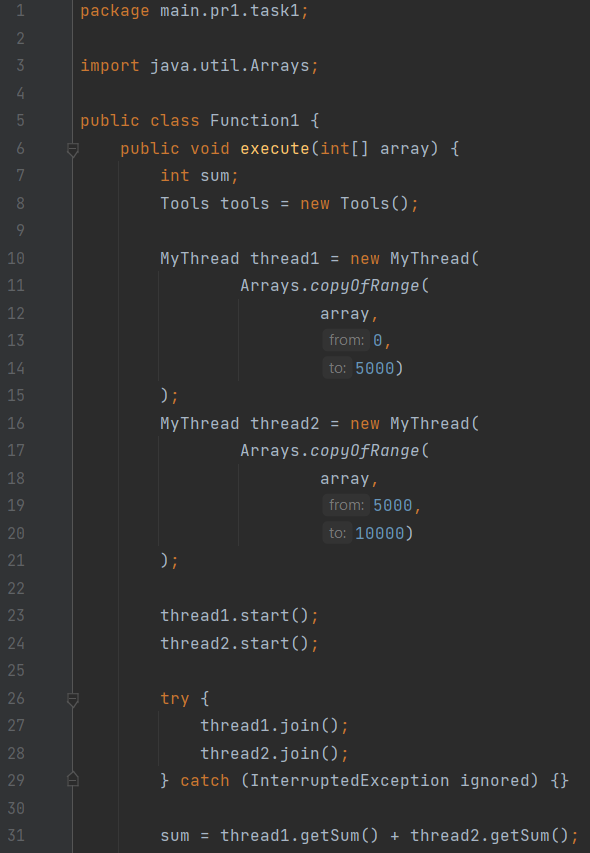


Рисунок 3 – Скриншот кода класса для реализации метода многопоточности (Часть 1)

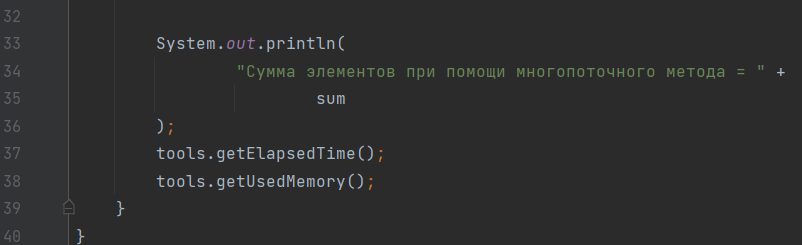


Рисунок 4 – Скриншот кода класса для реализации метода многопоточности (Часть 2)

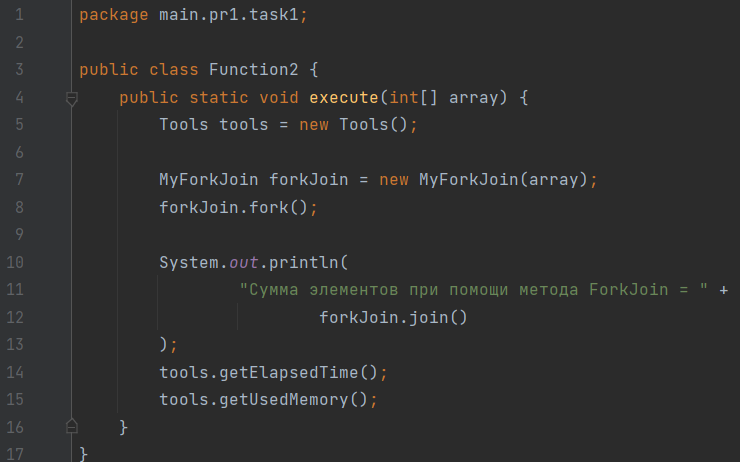


Рисунок 5 – Скриншот кода класса для реализации метода ForkJoin



Рисунок 6 – Скриншот кода класса наследника Thread

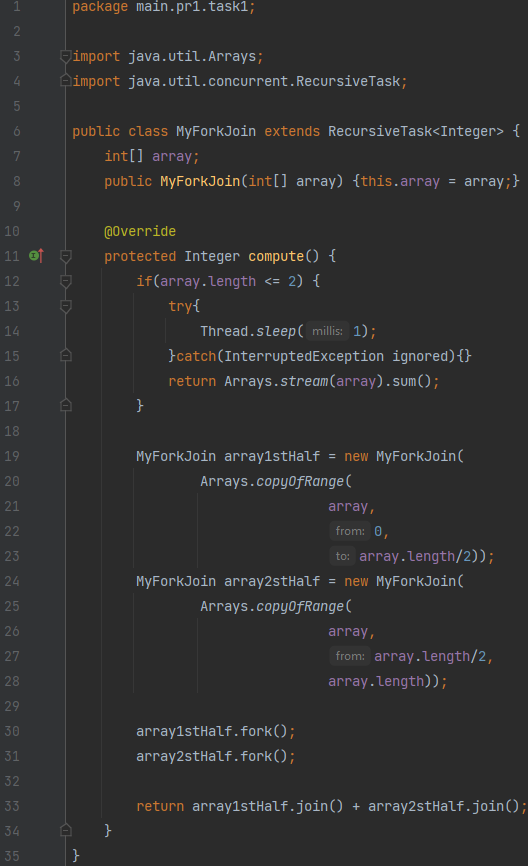


Рисунок 7 – Скриншот кода класса наследника для реализации метода ForkJoin



Рисунок 8 – Скриншот кода класса вспомогательных методов

После запуска первой программы и ожидания около одной минуты для всех методов выводятся результаты вычислений, что можно увидеть на следующем рисунке (Рисунок 9):

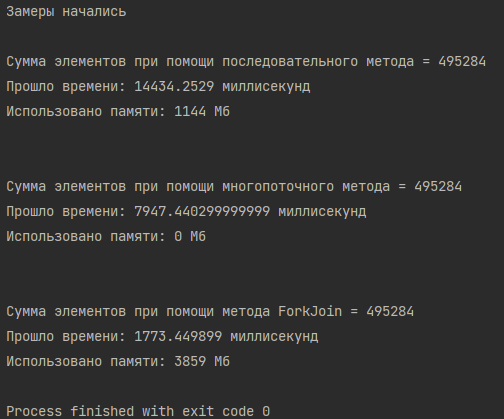


Рисунок 9 – Скриншот результата выполнения первой задачи

Затем производился запуск второй программы, в которой вводятся последовательно числа до получения результата вычислений, поскольку вычисления производятся с желаемой задержкой (1-5 секунд), результат выполнения задания представлен на рисунке 10.

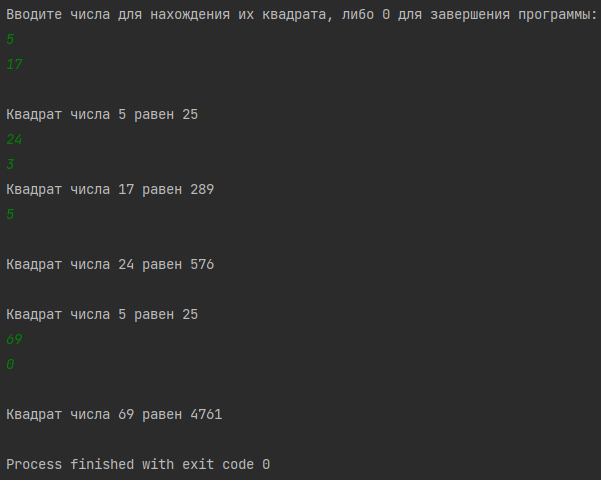


Рисунок 10 – Скриншот результата выполнения второй задачи

Наконец, запуск третьей программы производился с целью генерации и дальнейшей обработки ранее сгенерированных файлов в разных потоках, поскольку записей с новых строк больше, чем умещается на скриншоте, результат выполнения третьей задачи представлен на рисунках 11-12.

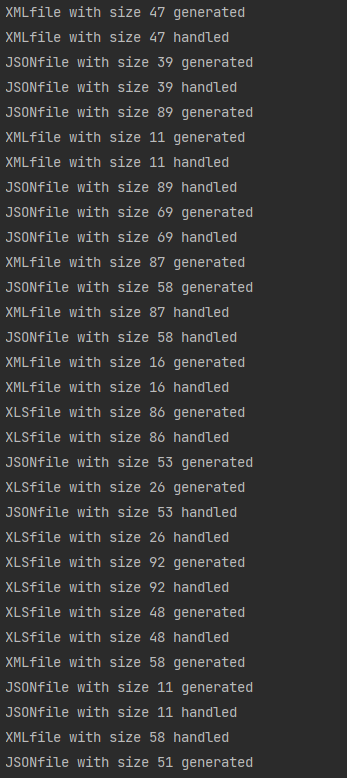


Рисунок 11 – Скриншот результата выполнения третьей задачи (Часть 1)

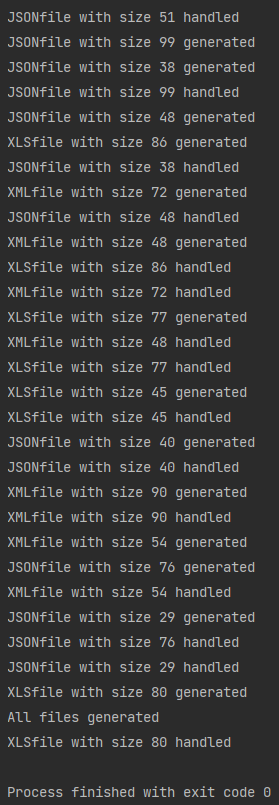


Рисунок 12 – Скриншот результата выполнения третьей задачи (Часть 2)

**Вывод:** в ходе выполнения практических работ по дисциплине «Разработка клиент-серверных приложений» были получены полезные теоретические, а главное практические навыки создания приложений, использования многопоточности разными методами (наследованием классов Thread и ForkJoin), создания и обработки файлов.