

Теория вычислительных процессов

Задания

Задание 1. Вычисления в отдельном потоке

В приложении с GUI реализовать:

- Запуск вычислений в основном потоке.
- Запуск вычислений в отдельном потоке. При этом окно программы должно быть доступно для взаимодействия с пользователем.
- *Дополнительно:*
 - *Показывать прогресс вычислений*

Примеры вычислительных задач:

- Вычисление числа π
- Операции с большими массивами: сортировка, нахождение суммы, ...
- Задачи их численных методов
- Обработка изображений
- Фракталы, построение множеств Мандельброта или Жулиа
- Генетический алгоритм, ...

Рекомендации к выполнению

Примеры работы с потоками: github.com/ivtipm/ProcessCalculus/tree/master/examples

В программе из этого задания поток с отдельными вычислениями должен быть запущен в методе – обработчике нажатия на кнопку. Сложность заключается в том, что нужно после завершения вычислений в отдельном потоке обновить данные в окне программы. Однако обработчики событий окна программы находятся в основном потоке.

Qt

В Qt для этого выгодно использовать механизм сигналов и слотов. Создать отдельный класс – для вычислений в отдельном потоке, вызвать в нём сигнал `finished()`. Соединив (`QObject::connect`) этот сигнал с обработчиком – методом на форме можно обновить данные, когда вычисления завершатся. Пример: github.com/ivtipm/ProcessCalculus/tree/master/examples/example_qthread

JavaFX

Создавая интерфейс пользователя с помощью JavaFX можно обернуть обращение к элементам интерфейса из другого потока в производный от `Runnable` класс следующим образом:

```
// запуск потока в одном из методов контролёра
в new Thread( () -> {
    // продолжительные вычисления
    Platform.runLater(() -> {
        // обновление Lable в окне программы
        status_label.setText( "Done!" );
    });
}).start();
```

Статический метод `runLater`¹ поставит переданный в него объект типа `Runnable` в очередь обработки сообщений программы.

1 <https://docs.oracle.com/javase/8/javafx/api/javafx/application/Platform.html#runLater-java.lang.Runnable>