# Задание

Разработать программу, содержащую 2 потока – производителя и потребителя. Для обоих потоков необходимо предусмотреть настройки скорости обработки (время ожидания, от 0 до 10 000 мс), приоритета выполнения, включения/выключения паузы.

При запуске программы оба потока должны находиться в режиме паузы.

## Дополнительное задание (необязательно)

Предусмотреть возможность создания произвольного числа производителей и потребителей.

# Рекомендуемая литература

Потоки в Delphi (Базовые сведения)

<http://www.delphi-manual.ru/threads.php>

<http://www.delphi-manual.ru/synchronize.php>

Подробная статья с описанием большинства «подводных камней»:

Martin Harvey.2000. Многопоточность - как это делается в Дельфи.

Перевод на русский язык: Борис Новгородов, 2002 <http://forum.vingrad.ru/forum/topic-60076/anchor-entry480384/0.html>

Справочник по COM и ActiveX в Delphi. Глава 3. Использование потоков в приложениях Windows

<http://it-ebooks.ru/load/guides/spravochnik_po_com_i_activex_v_delphi/3-1-0-224>

<http://bourabai.ru/alg/com/gl03.htm>

# Варианты

|  |  |
| --- | --- |
| Андреюшкина Т. |  |
| Беликова Н. |  |
| Валиев А. |  |
| Виноградов А. |  |
| Груднов Д. |  |
| Кандалова И. |  |
| Киселев М. |  |
| Крючкова В. |  |
| Кспоян Г. |  |
| Мишанин М. |  |
| Федорова И. |  |
| Чугунова Е. |  |

Производитель генерирует квадратные уравнения со случайными параметрами a, b, c. Потребитель извлекает одно уравнение из списка по методу LIFO (стек) и решает его. Если уравнение не имеет корней, то это считается отказом.

Показать на форме текущее состояние стека, все решенные уравнения и число отказов.

Производитель считывает текущее системное время, извлекает и него число минут (от 0 до 59) и добавляет в список. Потребитель извлекает из списка все имеющиеся в данный момент значения, вычисляет их среднее и очищает список.

Показать на форме текущее состояние списка, все записанные производителем числа и все вычисленные потребителем средние.

Производитель генерирует массу и радиус шара. Потребитель извлекает один шар из списка по методу FIFO (очередь) и находит его плотность.

Показать на форме текущее состояние очереди и параметры всех обработанных шаров.

Производитель генерирует случайный цвет в формате RGB и помещает в список. Потребитель извлекает из списка случайный (по номеру в списке) цвет. Если цвет светлый (сумма R+G+B больше, чем 3\*128), то он успешно потребляется и его цена = (R+G+B) / 3 прибавляется к общей сумме покупок. В противном случае цвет просто удаляется из стека (отказ от покупки).

Показать на форме текущее состояние списка, все сгенерированные производителем цвета, список успешно потребленных цветов, сумма покупок, число отказов.

Производитель генерирует три угла треугольника a, b, c – целые числа, в сумме дающие 180. Потребитель извлекает один треугольник из списка по методу FIFO (очередь) и определяет тип треугольника (остроугольный, прямоугольный, тупоугольный, равнобедренный, равносторонний).

Показать на форме текущее состояние очереди, список сгенерированных треугольников, тип каждого проверенного треугольника, число треугольников каждого типа.

Производитель генерирует случайный пол и наличие гена кучерявой шерсти у особей некоторого вида. Потребитель извлекает две случайных особи из списка (популяции). Если особи одного пола, то ситуация считается отказом, эти особи остаются в списке. Если особи разного пола, то они удаляются из популяции и скрещиваются, причем потомок будут кучерявыми, только если оба родителя кучерявые (ген кучерявости является рецессивным). Потомок в популяцию не возвращается.

Показать на форме текущее состояние популяции, список сгенерированных особей, список результатов скрещиваний, количество кучерявых и гладкошерстных потомков.

Производитель записывает текущее положение курсора на экране в список. Потребитель извлекает все записанные в список координаты и вычисляет среднее положение курсора.

Показать на форме текущее содержимое списка, все записанные производителем координаты, все вычисленные потребителем средние, количество замеров у потребителя и производителя.

Производитель генерирует эллипсы со случайными радиусами. Потребитель извлекает один эллипс из списка по методу LIFO (стек) и вычисляет его площадь.

Показать на форме текущее состояние стека, все сгенерированные и обработанные эллипсы и их количество.

Производитель генерирует случайные цены на товар (минимальную и максимальную цену задает пользователь). Потребитель извлекает одну случайную цену из списка и сравнивает со средней ценой по всем предыдущим попыткам. Если цена меньше или равна средней, или среднее еще не вычислялось (первая попытка), то потребитель «покупает товар» – удаляет его из списка, в противном случае товар остается в списке. Средняя цена пересчитывается и при удачной, и при неудачной покупке.

Показать на форме текущее состояние списка, все купленные товары, текущую среднюю цену.

Производитель генерирует случайные строки текста длиной от 5 до 10 символов. Потребитель извлекает первую строку из списка по методу LIFO (стек) и проверяет, есть ли в ней символы помимо цифр и латинских букв. Если посторонние символы есть, строка считается некорректной.

Показать на форме текущее состояние стека, список некорректных строк, общее количество корректных и некорректных строк.

Производитель генерирует точки на плоскости со случайными вещественными координатами (могут быть и отрицательными). Потребитель извлекает из списка две случайных точки. Та точка из пары, которая окажется дальше от начала координат, удаляется из списка.

Показать на форме текущее состояние списка, список всех сгенерированных и всех удаленных точек, а также координаты самой далекой и самой близкой к началу координат точек среди проверенных потребителем.

Производитель последовательно вычисляет значения ряда Фибоначчи, используя два верхних значения в списке. Потребитель удаляет случайное количество значений в вершине списка.

Показать на форме текущее состояние списка, количество и список удаленных чисел, а также максимальное найденное за все время число.