**Лабораторная работа №20**

*«LINQ»*

**Рекомендации по программированию**

* Поля в классе предпочтительнее делать private.
* Интерфейс должен быть полным, т.е. предоставлять возможность выполнять любые разумные действия с классом. И одновременно минимально необходимым, т.е. без дублирования и пересечения возможностей методов.
* Каждый метод класса должен решать только одну задачу. Не стоит объединять два коротких независимых фрагмента кода в один метод.
* Если действия встречаются хотя бы дважды, стоит оформить их в отдельные функции.
* Все заданные методы, поля и экземпляры классов должны иметь осмысленные имена.
* Каждое имя интерфейса должно начинаться с буквы I.

**Требования к лабораторной**

* Каждый разрабатываемый класс должен, как правило, содержать следующие элементы: поля с различными спецификаторами, конструкторы с параметрами и без параметров, методы, свойства.
* Методы и свойства должны обеспечивать непротиворечивый, полный, минимальный и удобный интерфейс класса.
* При возникновении ошибок должны выбрасываться исключения
* Каждый студент должен реализовать свой класс из индивидуального задания со своими полями и методами. Любое совпадение не случайно и карается по закону джунглей.
* Проделанную работу необходимо показать на паре.

**Вариант 1**

Создать приложение содержащих два порожденных потока. Первый поток производит добавление в начало файла новой строки, второй поток удаление последней строки. При невозможности удаления строки поток должен приостанавливаться на время необходимое первому потоку для добавления новой строки.

**Вариант 2**

Винни-Пух и пчелы. Заданное количество пчел добывают мед равными порциями, задерживаясь в пути на случайное время. Винни-Пух потребляет мед порциями заданной величины за заданное время и столько же времени может прожить без питания. Работа каждой пчелы реализуется в порожденном потоке.

**Вариант 3**

Поиск всех простых чисел (простым называется число, которое является своим наибольшим делителем) в указанном интервале чисел, разделенном на несколько диапазонов. Обработка каждого диапазона производится в порожденном потоке. Классический алгоритм Евклида определения наибольшего общего делителя двух целых чисел (x, y) может применяться при следующих условиях:

* оба числа x и y неотрицательные;
* оба числа x и y отличны от нуля.

На каждом шаге алгоритма выполняются сравнения:

* если x == y, то ответ найден;
* если x < y, то y заменяется значением y - x;
* если x > y, то x заменяется значением x - y.

**Вариант 4**

Создать приложение выполняющее перемещение символа. Первый поток производит перемещение символа согласно текущих значений приращения х, у. Второй поток производит расчет текущего значения приращения х,у. У пользователя должна быть возможность менять правило расчета х,у – выбрав из выпадающего списка определенный элемент. Выпадающий список содержит элементы – “не перемещаться”, "по прямой", "sin(x)", "cos(x)".

**Вариант 5**

Умножение матрицы на вектор, умножение матрицы на матрицу. Обработка одной строки матрицы - в порожденном потоке.

**Вариант 6**

Шарики. Координаты заданного количества шариков изменяются на случайную величину по вертикали и горизонтали. При выпадении шарика за нижнюю границу допустимой области шарик исчезает. Изменение координат каждого шарика в отдельном потоке.

**Вариант 7**

Противостояние нескольких команд. Каждая команда увеличивается на случайное количество бойцов и убивает случайное количество бойцов участника. Борьба каждой команды реализуется в отдельном потоке.

**Вариант 8**

Создайте приложение, в отдельном потоке вычисляющее значение w и непрерывно обновляющего его. Для расчета числа использовать следующую формулу.

W = 1 + sin(x) – 2\*cos(x) + 4\*sin2(x) – 8\*cos2(x) +..

**Вариант 9**

Авиаразведка. Создается условная карта в виде матрицы, размерность которой определяет размер карты, содержащей произвольное количество единиц (целей) в произвольных ячейках. Из произвольной точки карты стартуют несколько разведчиков (потоков), курсы которых выбираются так, чтобы покрыть максимальную площадь карты. Каждый разведчик фиксирует цели, чьи координаты совпадают с его координатами и по достижении границ карты сообщает количество обнаруженных целей.

**Вариант 10**

Бег с препятствиями. Создается условная карта трассы в виде матрицы, ширина которой соответствует количеству бегунов, а высота – фиксирована, содержащей произвольное количество единиц (препятствий) в произвольных ячейках. Стартующие бегуны (потоки) перемещаются по трассе и при встрече с препятствием задерживаются на фиксированное время. По достижении финиша бегуны сообщают свой номер.

**Вариант 11**

Создать поток, который управляет перемещением символа. Символ движется по Х. Создать 3 символа и устроить "тараканьи" бега среди них.

**Вариант 12**

Параллельно считать значения функций из заданного диапазона для х [a,b]: y = sin(x), y = 4\*x2-2\*x – 22, y = ln(x2)/x3.

**Вариант 13**

Однорукий бандит - 3 потока, генерирующие числа от 0 до 9. По нажатию клавиши потоки останавливаются и результат анализируется. При анализе использовать следующие комбинации (три одинаковых числа, два одинаковых числа, три единицы, три семерки, две единицы, имеется четверка)

**Вариант 14**

Параллельно вычислить 10-ю степень двойки 1 - сложением, умножением и просто возведением в степень.

**Вариант 15**

Создать два потока. Первый поток производит запись в файл случайных данных. Второй производит чтение данных из этого файла и вывод их на экран.

**Вариант 16**

Создать два потока. Первый ищет числа Фибоначчи (каждое последующее число равно сумме двух предыдущих чисел), второй простые числа. Результат работы каждого потока сохраняются в отдельный файл. После остановки потока – программа производит анализ файлов, выводит их на экран, а также показывает количество найденных чисел Фибоначчи и простых чисел.

**Вариант 17**

Создать игру, где будут 2-3 барана и волк. При совпадении координат волка с бараном баран исчезает. При совпадении координат баранов появляется новый баран. Все движутся хаотически

**Вариант 18**

Создать приложение вычисляющее значение функции y = 23\*x2 – 33, с шагом x=0.01 из диапазона, вводимого с клавиатуры [a,b]. Первый поток производит расчет данных функции и добавляет их в конец массива данных. Второй поток извлекает из массива данных значения и производит выводит функции на экран.

**Вариант 19**

Создайте приложение, в отдельном потоке вычисляющее значение π и непрерывно обновляющего его в пользовательском интерфейсе. Для расчета числа использовать следующую формулу.



**Вариант 20**

Создать приложение выполняющее сортировку массива данных и визуального отображения процесса сортировки на экране. Первый поток производит сортировку по возрастанию, второй по убыванию. После каждого перемещения элемента результирующий производится вывод текущего состояния сортировки. Каждый поток работает с отдельным экземпляром массива данных. Состояние сортировки выводится в двух элементах

**Вариант 21**

Создать приложение выполняющее сортировку массива данных и визуального отображения процесса сортировки на экране. Первый массив производит сортировку по возрастанию, второй по убыванию, третий производит вывод массива на экран. Каждый поток работает с общим экземпляром массива данных. Состояние сортировки выводится в одном элементе.

**Вариант 22**

Warcraft. Заданное количество юнитов добывают золото равными порциями из одной шахты, задерживаясь в пути на случайное время, до ее истощения. Работа каждого юнита реализуется в порожденном процессе потоке.

**Вариант 23**

Контрольная сумма. Для нескольких файлов (разного размера) требуется вычислить контрольную сумму (сумму кодов всех символов файла). Обработка каждого файла выполняется в отдельном потоке.

**Вариант 24**

Поиск указанной с клавиатуры строки в файле, содержавшим первую часть первого тома «Война и мир». Обработка одной строки в порожденном потоке.