Домашнее задание №4

## Общие замечания.

* При решении задач нужно использовать только те возможности языка Go, которые были рассмотрены на тренинге на момент выдачи домашнего задания (некоторые исключения могут оговариваться особо в условии конкретных задач).
* Решение каждой задачи помещайте в отдельный каталог вида Task\_NN, например – Task\_01 (см. примеры кода к лекциям).
* Все решения запакуйте в один zip-архив и отправьте по почте тренеру.

## Задача 1.

Ваша задача – создать обобщённый тип для Стека. Используйте реализацию стека с помощью **односвязного списка**. Снабдите тип для стека методами Push, Pop и IsEmpty. Сделайте так, чтобы метод Pop возвращал пару – «значение-ошибка». Тип ошибки – стандартный интерфейс error (используйте пакет errors). Ошибка возвращается, когда мы пытаемся извлечь элемент из пустого стека.

## Задача 2.

*Разреженные числовые матрицы* – числовые матрицы, в которых только относительно малая часть элементов не равна нулю (<https://en.wikipedia.org/wiki/Sparse_matrix>). Например, если в матрице размером 100x200 только угловые элементы ненулевые (их 4 штуки), то мы определённо можем назвать такую матрицу разреженной. Очевидно, что в целях экономии памяти нулевые элементы разреженной матрицы желательно не хранить.

Необходимо создать обобщённый тип для представления разреженной числовой матрицы. Элементами матрицы могут быть стандартные числовые типы (кроме комплексных чисел) и типы, созданные на их основе. Предусмотрите методы для чтения и записи элемента матрицы (элемент определяется двумя индексами). В этой задаче при некорректности значений (например, индексы выходят за допустимый диапазон) вызывайте панику.

## Задача 3.

Реализуйте функцию where (см. ДЗ №2, задача 2) в виде обобщённой функции с распараллеливанием. Запустите восемь горутин, каждая из которых фильтрует свою часть исходного среза. Результаты фильтрации затем объединяются в итоговый срез.

Важно: результаты работы обычной функции where и функции where с горутинами могут отличаться! Это будут срезы из одних и тех же элементов, но взаимный порядок этих элементов может быть другим.

Проведите замеры производительности – есть ли выгода от использования горутин? При каком размере среза-аргумента такая выгода начинает проявляться (на вашем компьютере)?