



Художник Мария Усеинова

Очередь – ИЗ СТЕКОВ

Света и её старший брат программист Лёша ужинали. Брат поинтересовался, как прошёл день.

– Когда шли обедать, ехали на лифте, и я заметила закономерность. Почти на каждом этаже мы продвигались вглубь лифта, чтобы другие могли зайти. А потом зашедшие позже оказались в очереди в столовой впереди тех, кто в лифт зашёл раньше. Несправедливо!

– Интересное правило получилось! Напоминает одну структуру данных из программирования: *стек*.

– А что это значит?

– В стеке программа может хранить данные. Она может добавлять туда новый элемент с помощью операции *push*. И ещё она может считывать элемент, который был туда добавлен последним, удаляя его. Это делает операция *pop*, а правило, по которому она действует, называют сокращённо *LIFO*, от английского *Last In, First Out*: последним пришёл – первым ушёл.

– То есть в стек мы записываем данные, но когда читаем из него, они возвращаются в обратном порядке?

– Именно так.

– А если мы захотим читать в том же порядке?

– Такая структура называется *очередь*. Она похожа на обычную очередь в столовую, а её правило называют сокращённо *FIFO*, от английского *First In, First Out*. У неё тоже есть операции добавления и считывания *push* и *pop*. Операция *push* работает так же, как у стека, а операция *pop* считывает элемент очереди, который был добавлен раньше всех, удаляя его. Хочешь задачку? Представь, что тебе поступают элементы в некотором порядке. У тебя есть только стеки и их операции *push* и *pop*. Придумай, как хранить эти элементы и выдавать их в порядке поступления.

– То есть просто использовать очередь нельзя?

– Да, нельзя. Нужно сконструировать структуру данных, которая ведёт себя как очередь.

– А сколько стеков можно использовать?

– Попробуй обойтись всего двумя!

Света тут же принялась сооружать на тарелке воображаемые стеки-лифты из макарон с фасолинами.

– Я подумаю, а ты не подсказывай.

Ответ в следующем номере