

## Вариант 4

### Задача 1

Напишите функцию `filterNestedProperty`, которая принимает массив объектов и строку в формате `"property.subproperty"`, возвращая массив объектов, где `subproperty` имеет значение `true`.

Например:

```
const data = [ { name: 'item1', details: { active: true } }, { name: 'item2', details: { active: false } }, ];
```

*Корректная фильтрация объектов с вложенным свойством — 15 баллов.*

### Задача 2

Напишите функцию `averageProperty`, которая принимает массив объектов и имя числового свойства, вычисляя среднее значение.

Например:

```
const items = [{ score: 20 }, { score: 30 }, { score: 40 }];
```

*Корректный подсчет среднего значения — 20 баллов.*

### Задача 3

Напишите функцию `fetchWithRetryAndTimeout`, принимающую URL, время ожидания и попытки. Если запрос не завершен за время ожидания, повторите его указанное число раз.

*Корректная обработка времени и повторные попытки — 25 баллов.*

### Задача 4

Реализуйте функцию `loadSelectedData`, принимающую массив URL и загружающую данные только для успешных URL.

*Корректная фильтрация успешных данных — 20 баллов.*

### Задача 5

Реализуйте асинхронный генератор `fetchPagesWithStopCondition`, прекращающий выполнение, если данные не загружены или достигнут максимум.

*Корректное выполнение и условия завершения — 20 баллов.*