Задание6_3. Шаблон Paginator

В методическом материале мы разработали функцию Head, которая позволяет пройтись циклом for по началу контейнера (стр. 14-20). В этой задаче мы сделаем шаг вперёд и разработаем шаблон Paginator, который разбивает содержимое контейнера на несколько страниц. Классический пример, когда такое может пригодиться на практике, — это распределение списка мобильных приложений по экранам телефона. Допустим, у нас есть вектор всех приложений нашего телефона и на одном экране мы можем разместить 20 иконок приложений. Тогда распределить приложения по экранам мы можем вот таким кодом:

```
vector<vector<Application>> DistributeAmongScreens(const vector<Application>& apps) {
vector<vector<Application>> result;
for (const auto& page : Paginate(apps, 20)) {
result.push_back({page.begin(), page.end()});
}
// result[0] - все приложения, которые попадают на первый экран,
// result[1] - все приложения, которые попадают на второй экран и т.д.
return result;
}
```

Заметьте, наш код получился коротким и элегантным. Нам не пришлось писать какой-то отдельный код для обработки последнего экрана, который может содержать меньше 20 приложений.

Итак, разработайте шаблон класса Paginator со следующими свойствами:

- он имеет один шаблонный параметр — тип итератора
- конструктор класса Paginator<Iterator> принимает три параметра:
    1. Iterator begin
    2. Iterator end — пара итераторов begin и end задают полуинтервал [begin; end), который мы будем нарезать на страницы
    3. size_t page_size — размер одной страницы
- по объектам класса Paginator<Iterator> можно проитерироваться с помощью цикла range-based for
- класс Paginator<Iterator> имеет метод size_t size() const, который возвращает количество страниц, на которые был разбит переданный контейнер
- сами страницы должны так же поддерживать итерацию с помощью range-based for и иметь метод size_t size() const, возвращающий количество объектов в этой странице
- подробные примеры использования смотрите в юнит-тестах в шаблоне решения

Кроме того разработайте шаблон функции Paginate, которая принимает ссылку на контейнер и размер страницы, и возвращает объект класса Paginator<It>:

```
template <typename C> ??? Paginate(C& c, size_t page_size)
```

Файл с заготовкой решения задачи

paginator.cpp

Возможно, в шаблоне у вас будет некорректно работать вызов ASSERT_EQUAL в функции TestLooping. Разберитесь, почему это происходит, внесите правку в свою локальную версию файла test_runner.h, чтобы подобная ошибка не возникала в других задачах.