

Занятие в терминал-классе № 4 от 29.09.2022

Последовательность действий:

1. Создать новый репозиторий (ЕСЛИ ХОТИТЕ НАЧАТЬ С ЧИСТОГО ЛИСТА НОРМАЛЬНО, если до этого не получалось), положить в папку файл .gitignor (про данный файл доп. инструкция в конце), прикрепить репозиторий к новой пустой папке - ПКМ git bush here:

ЕСЛИ ВСЁ ИТАК БЫЛО ОК И НА ГИТХАБЕ ВСЁ ЕСТЬ: клонировать существующий репозиторий.

Перейти в рабочую ветку (либо создать, либо в уже существующую).

2. Подготовка для написания кода.

ЕСЛИ С НУЛЯ: Создать новое Решение (название TC_continuation), добавить туда один Проект (название ArraysActions_Easy), один файл (arrays_easy.c).

ЕСЛИ КЛОНИРОВАЛИ РЕПОЗИТОРИЙ: открыть .sln. Добавить в решение один Проект (название ArraysActions_Easy), один файл (arrays_easy.c).

Написать шаблон для кода в arrays_easy.c. **Выложить на гитхаб эту заготовку.**

3. Реализовать задачу (см. далее). Каждая подзадача в ней - коммит. Учимся формировать историю коммитов.

Примерная история коммитов:

1. Создал и проинициализировал массив, вывел его на экран.
2. Добавил использование алгоритма «удаления» заданных элементов.
3. Пофиксил стиль кодирования.

Могут быть коммиты «исправил баг с ...»

Постановка задачи

1. Создать массив. Предложить пользователю ввести размер массива. Сгенерировать элементы случайно (от -10 до 10). Вывести массив на экран.
2. Все элементы, кратные заданному пользователем числу, возвести в заданную пользователем степень от 2 до 5. Вывести массив в обратном порядке.
3. Развернуть массив на месте (не напечатать в обратном порядке, а именно изменить порядок элементов). Напечатать массив на экран.
4. «Удалить» элементы меньше числа, заданного пользователем.

Как формировать .gitignore файл самостоятельно

1. Создаём файл .gitignore

2. Добавляем туда все папки или типы файлов, которые не должны оказаться на гитхаб! Для нас это как минимум папки формируемые при отладке или запуске релизной версии кода.

```
debug/  
*/debug/  
**/debug/  
  
release/  
*/release/  
**/release/  
  
x64/  
*/x64/  
**/x64/  
  
x86/  
*/x86/  
**/x86/  
  
.vs/  
*/.vs/  
**/.vs/
```

Команды, если есть трудности с гитхаб

1.

```
git init  
git remote add origin <ссылка на новый репозиторий>  
git checkout -b <название ветки>
```

или

```
git clone <ссылка на репозиторий>  
git checkout -b <название ветки>
```

2.

```
git add .  
git commit -m "добавлена заготовка для проекта ArraysActions_Easy"  
git push origin <название рабочей ветки>
```

3.

Постоянное сочетание команд

```
git add .  
git commit -m "бла бла бла"
```

А после финального коммита

```
git push origin <название рабочей ветки>
```

