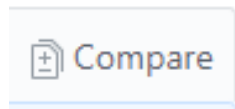


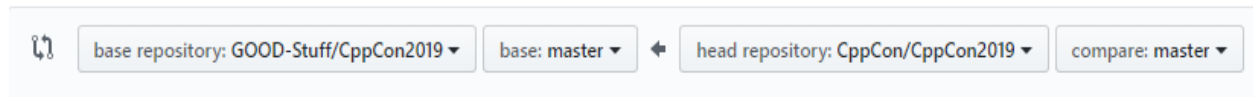
Лабораторное занятие №4

Густов Владимир Владимирович
gutstuf@gmail.com

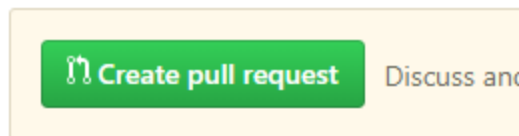
Обновление репозитория



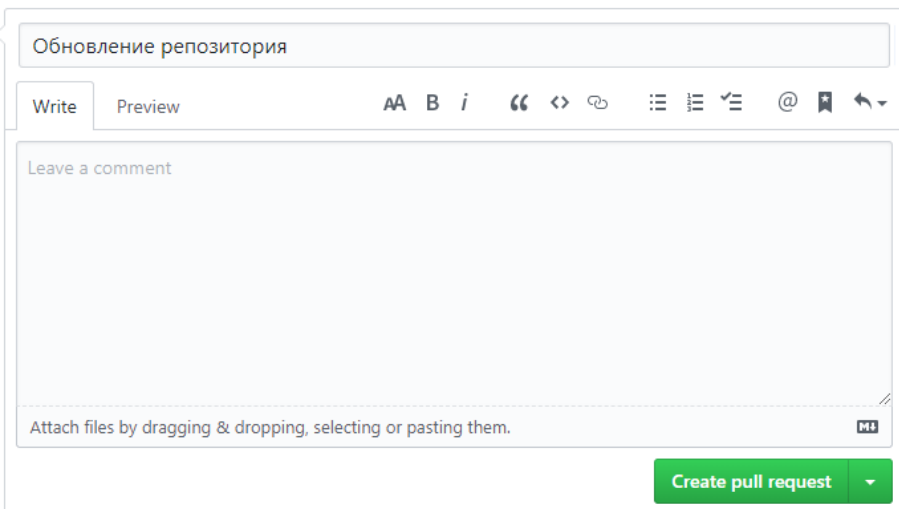
В своём форке нажимаем на кнопку «Сравнить»



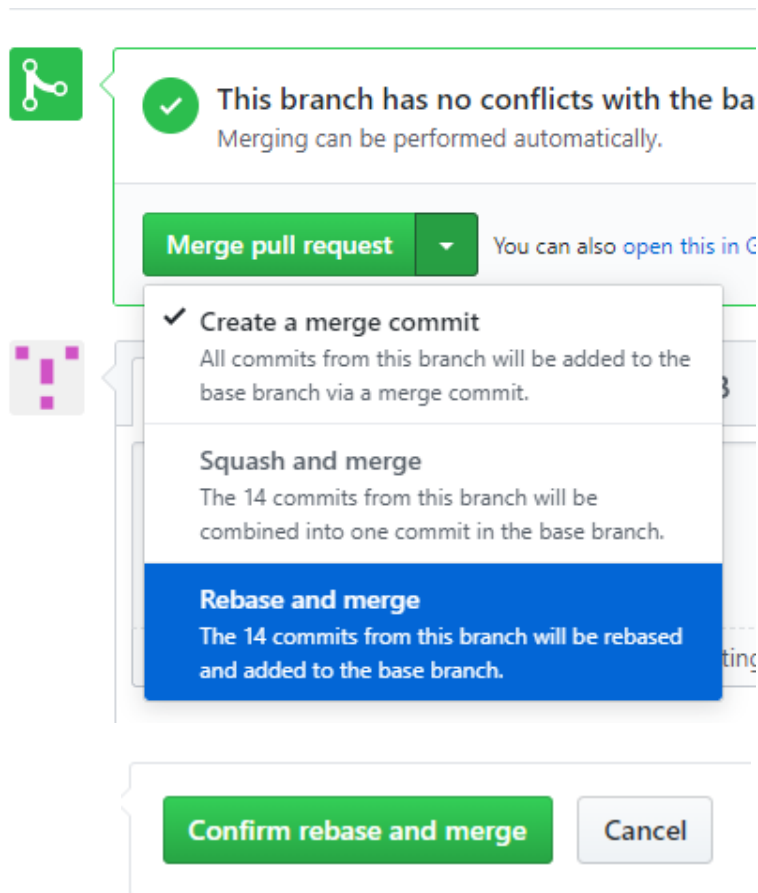
Выставите сравнение так, чтобы **base repository** являлся **вашим** форком, а **head repository** моим репозиторием (или нажмите на надпись «switching the base»)



Нажмите кнопку «Создать pull request»



Добавьте описание создающегося запроса и нажмите на кнопку «Создать pull request»



Ниже, в диалоговом окне, выберите «**Rebase and merge**» и нажмите на кнопку

Подтвердите слияние нажав на кнопку «**Confirm rebase and merge**»

???

PROFIT

Словарь (таблица)

- представляет из себя ассоциативный массив (ключ + значение);
- реализуется с помощью сбалансированного бинарного дерева поиска (AVL, КЧД, etc);
- не содержит дубликаты ключей (для этого есть мультисловарь);
- значения могут дублироваться, например:

```
map<string, string> m;  
m.emplace(std::make_pair("Это ключ", "это значение"));  
m.emplace(std::make_pair("Новый ключ", "это значение"));  
m.emplace(std::make_pair("Это ключ", "а это новое значение"));  
// последняя вставка не будет исполнена, т.к. ключ дублируется  
for (auto& el: m)  
    cout << el.first << " " << el.second << endl;
```

Вывод:

Это ключ это значение

Новый ключ это значение

Хеш-таблица

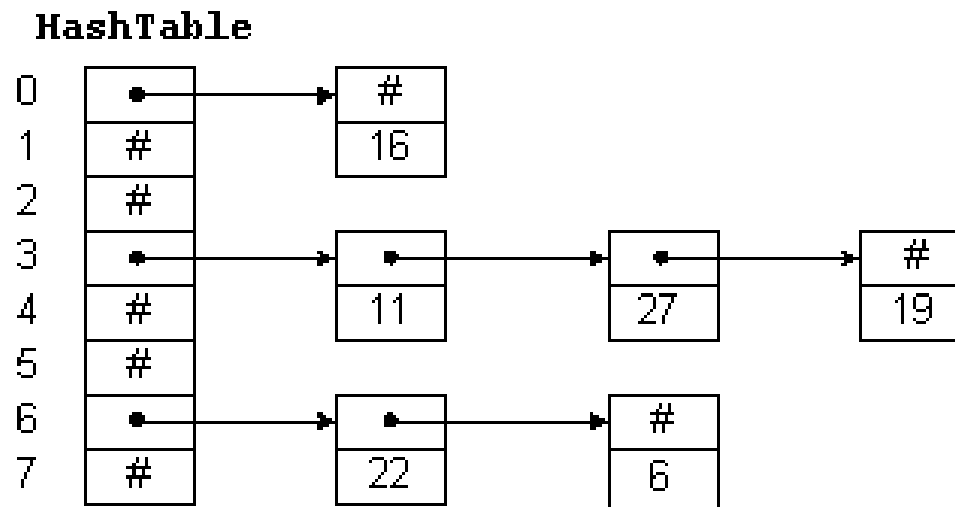
Хеш (хеш-сумма) – результат обработки данных хеш-функцией (CRC32, SHA, MD5, etc).

Хеширование (хеш-функция) – преобразование данных произвольной длины в битовую строку установленной длины.

Хеш-таблица – абстрактная структура данных, реализующая ассоциативный массив. Использует хеширование для вычисления индекса элемента (хеш).

Коллизия – случай, при котором хеш-функция выдаёт одинаковый хеш для разных входных данных. Вероятность возникновения коллизий используется для оценки качества хеш-функций.

Для решения коллизий используется метод цепочек, либо метод открытой адресации.



Пример простейшей (но плохой, с точки зрения коллизий) хеш-функции:

```
int calc_hash(string key) {  
    int hash = 0;  
    for(auto el: key)  
        hash += el ^ hash;  
    return hash;  
}
```

<http://www.cse.yorku.ca/~oz/hash.html> - ещё несколько вариантов хеш-функций (имхо, не самых надёжных)

В общем:

- является ассоциативным массивом (ключ + значение);
- реализуется с помощью массива списков
- исходя из свойств словаря (БДП), элементы хранятся будучи отсортированными;
- у хеш-таблицы элементы не отсортированы;