

Home task

1. Частичная специализация шаблона (partial template specialization)

Необходимо воспользоваться заготовкой ht1.cpp на гитхабе в папке HW

```
template<class T, size_t N>
class Container{
private:
    T m_array[N];
public:

    Container(){
        for(int i = 0 ; i < N ; i ++){
            m_array[i] = (T)0;
        }
    }

    void print(){
        for(const auto & a: m_array)
        {
            std::cout<<a<<"\t";
        }
        std::cout<<"\n";
    }

    void fill(const T array[N])
    {
        std::copy(array,array + N , m_array);
    }
};
```

Это шаблонный класс с двумя параметрами – параметр класса и non-type параметр. Необходимо частично его специализировать:

1.1.

```

class Item
{
private:
    int * arr;
public:
    Item();
    Item(size_t size); //необходимо выделить сюда массив размера size и заполнить его случайными числами до 100
    //как делать --- см rand()
    friend std::ostream & operator<<(std::ostream& out, const Item & it );
    double average(); //не забываем о приведении типов.
    ~Item();
};

template<size_t N>
class Container<std::vector<Item>, N >
{
private:
    std::vector<int> m_vec;
public:
    Container();
    void print();
};

```

Необходимо таким образом специализировать этот класс, чтобы в конструкторе резервировалась память ровно под N объектов. (у вектора есть метод `reserve()`). А функция `print()` должна печатать среднее арифметическое для каждого объекта `item` внутри этого вектора.

1.2.

```

template<size_t N>
class Container<std::list<std::string>, N>
{
private:
    std::list<std::string> listOfStr;
public:
    Container();
    void print();
};

```

Реализовать специализацию для такого класса. К конструктору те же требования что и в 1.1., функция `print` должна печатать для каждого объекта количество запятых.

!!!Все объявленные функции необходимо определить, даже если они не применяются. Внутри класса или вне – решать вам!!!

2. STL containers

На следующем занятии будет тест(несложный) по последовательным контейнерам (sequence containers). Поэтому вам необходимо будет отработать и запомнить как устроены

-вставка

-удаление

-обход (по итератору в том числе)
для контейнеров : list, vector, array, deque.