

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ имени Н.Э.БАУМАНА
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)



Факультет: Информатика и системы управления

Кафедра: Теоретическая информатика и компьютерные технологии

Лабораторная работа №9
Реализация классов на языке C++
Вариант №27

Выполнил
студент группы ИУ9-21Б
Лисов Алексей

Москва, 2023

1 Условие

Приобретение навыков разработки приложений с использованием принципов ООП на C++. В варианте 27 необходимо было реализовать класс `SparseArray<T>`. `SparseArray<T>` – разреженный массив, отображающий неотрицательные целые числа в значения типа `T`. Массив должен быть реализован через хэш-таблицу. Требование к типу `T`: наличие конструктора по умолчанию (т. к. разреженный массив должен уметь создавать значения типа `T`). Условие задачи, исходный код и пример работы программы необходимо прислать в формате `LaTeX`.

2 Код решения

Файл `main.cpp`

```
#include <iostream>
#include <unordered_map>

using namespace std;

template<typename T>
class SparseArray {
public:
    SparseArray(const T& defaultValue = T()) : defaultValue(defaultValue) {}

    T& operator[](const int index) {
        if (map.find(index) == map.end()) {
            map[index] = defaultValue;
        }
        return map[index];
    }

    SparseArray<T> operator()(const int start, const int end) const {
        SparseArray<T> subarray(defaultValue);
        for (const auto& [index, value] : map) {
            if (index >= start && index < end) {
                subarray[index] = value;
            }
        }
        return subarray;
    }
}
```

```

    bool operator==(const SparseArray<T>& other) const {
        if (this->map.size() != other.map.size()) {
            return false;
        }
        for (const auto& [index, value] : map) {
            if (other.map.find(index) == other.map.end() || other.map.find(index)->second != value) {
                return false;
            }
        }
        return true;
    }

    bool operator!=(const SparseArray<T>& other) const {
        return !(*this == other);
    }

private:
    std::unordered_map<int, T> map;
    T defaultValue;
};

int main() {
    SparseArray<int> arr(0);

    arr[2] = 2;
    arr[4] = 4;
    arr[6] = 6;
    arr[8] = 8;

    cout << arr[8] << "\n";

    SparseArray<int> subarray = arr(2, 7);

    for (int i = 2; i < 7; i++) {
        std::cout << subarray[i] << " ";
    }
    cout << "\n";

    SparseArray<int> brr(0);

```

```
brr[2] = 2;
brr[4] = 4;
brr[6] = 6;
brr[8] = 8;

cout << (brr == arr) << "\n";

brr[7] = 7;

cout << (brr == arr) << "\n";
cout << (brr != arr);

return 0;
}
```

3 Скриншоты

```
SparseArray<int> brr( defaultValue: 0);  
  
brr[2] = 2;  
brr[4] = 4;  
brr[6] = 6;  
brr[8] = 8;  
  
cout << (brr == arr) << "\n";  
  
brr[7] = 7;  
  
cout << (brr == arr) << "\n";  
cout << (brr != arr);  
  
return 0;  
}
```

Рис. 1: Вывод программы

```

#include <iostream>
#include <unordered_map>

using namespace std;

template<typename T>
class SparseArray {
public:
    SparseArray(const T& defaultValue : T = T()) : defaultValue(defaultValue) {}

    T& operator[](const int index) {
        if (map.find(index) == map.end()) {
            map[index] = defaultValue;
        }
        return map[index];
    }

    SparseArray<T> operator()(const int start, const int end) const {
        SparseArray<T> subarray(defaultValue);
        for (const auto& [index, value] : map) {
            if (index >= start && index < end) {
                subarray[index] = value;
            }
        }
        return subarray;
    }
};

```

Рис. 2: Вывод программы

```

8
2 0 4 0 6
1
0
1
Process finished with exit code 0

```

Рис. 3: Вывод программы