**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ.УЛЬЯНОВА (ЛЕНИНА)**

Кафедра вычислительной техники

Отчет по лабораторной работе №2 на тему:

«Множество как объект»

по дисциплине «Алгоритмы и структуры данных»

Колинько П.Г.

Санкт-Петербург

2018

Торопов В. А.

Преподаватель

Студент гр. 7307

Оглавление

[Цель работы 3](#_Toc527985075)

[Задание 3](#_Toc527985076)

[Измерения времени 4](#_Toc527985077)

[Результат эксперимента с отслеживанием вызова функций-членов. 4](#_Toc527985078)

[Вывод 6](#_Toc527985079)

[Код программы 7](#_Toc527985080)

# **Цель работы**

Исследование эффекта от использования классов.

# **Задание**

Вычислить множество, содержащее цифры, имеющиеся в любом из множеств А, B, C, D.

# **Измерения времени**

|  |  |  |  |
| --- | --- | --- | --- |
|  | N = 1000 | N = 5000 | N = 10 000 |
| Массив /  ср. мощность | 524  8 | 2066  9 | 3654  8 |
| Список  ср. мощность | 1578  6 | 6231  7 | 14969  7 |
| Массив битов  ср. мощность | 531  15 | 1266  15 | 2750  15 |
| Машинное слово  ср. мощность | 0  6 | 0  7 | 16/100  8 |

\*N – кол-во измерений

# **Результат эксперимента с отслеживанием вызова функций-членов (Вариант «списки»)**

Create A

Create B

Create C

Create D

Create E

generated: A = [DCBA5]

A = [DCBA5]

generated: B = [ECB9643210]

B = [ECB9643210]

generated: C = [EDC9765320]

C = [EDC9765320]

generated: D = [EDCBA7420]

D = [EDCBA7420]

operator | got: A = [DCBA5]

& B = [ECB9643210]

Operator |= got: A = [DCBA5]

& B = [ECB9643210]

return: A = [0123469EDCBA5]

delete: A

operator | got: A = [0123469EDCBA5]

& C = [EDC9765320]

Operator |= got: A = [0123469EDCBA5]

& C = [EDC9765320]

return: A = [70123469EDCBA5]

delete: A

operator | got: A = [70123469EDCBA5]

& D = [EDCBA7420]

Operator |= got: A = [70123469EDCBA5]

& D = [EDCBA7420]

return: A = [70123469EDCBA5]

delete: A

operator = got: E = []

& A = [70123469EDCBA5]

return: R = [5ABCDE96432107]

delete: A

R = [5ABCDE96432107]

Middle power = 7

Time = 16/1

delete: R

delete: D

delete: C

delete: B

delete: A

Process finished with exit code 0

# Вывод

В ходе данной работы были получены практические знания по использованию классов. Они имеют ряд преимуществ: компактное расположение данных, открытые и закрытые части класса позволяют свести количество аргументов функций-членов к минимуму. Программа реализованная с использованием классов по времени выполнения не проигрывает программе, реализованной без классов.

# **Код программы**

Main:

#include <iostream>  
#include "Set.h"  
  
**const long** q0 = 1;  
  
**int** main() {  
 srand(time(**nullptr**));  
 Set A('A'), B('B'), C('C'), D('D'), E('E');  
 A.generator();  
 A.show();  
 B.generator();  
 B.show();  
 C.generator();  
 C.show();  
 D.generator();  
 D.show();  
 clock\_t t1 = clock();  
 **for**(**long** q = 0; q < q0; q++) {  
 E = A | B | C | D;  
 }  
 clock\_t t2 = clock();  
 E.show();  
 cout << "Middle power = " << (A.power() + B.power() + C.power() + D.power() + E.power()) / 5 << endl;  
 cout << "Time = " << t2 - t1 << "/" << q0 << endl;  
 **return** 0;  
}

Set.h:

#ifndef INC\_2\_SET\_H  
#define INC\_2\_SET\_H  
#include <iostream>  
#include <time.h>  
#endif //INC\_2\_SET\_H  
  
**using namespace** std;  
  
**class** Set{  
**private**:  
 **static int** N;  
 **int** n;  
 **char** S, \*A;  
**public**:  
 Set **operator** | (**const** Set&) **const**;  
 Set & **operator** |= (**const** Set&);  
 Set & **operator** = (**const** Set&);  
 **void** show();  
 **int** power() {**return** n;};  
 Set(**char** S1 = '\0'){  
 n = 0;  
 A = **nullptr**;  
 S = S1;  
 }  
 **void** generator(){  
 **int** wA = 0;  
 wA = rand() % 0xFFFF;  
 A = **new char** (N);  
 **for** (**int** i = 0; i < N - 1; i++){  
 **if** ((wA >> i) & 1){  
 A[n++] = (**char**)(i <= 9 ? i + '0' : i + 'A' - 10);  
 }  
 }  
 A[n] = '\0';  
 }  
 ~Set(){**delete** [] A;}  
};  
  
**int** Set :: N = 16;  
  
Set & Set::**operator** |=(**const** Set & B){  
 **for**(**int** i = 0; i < B.n; ++i){  
 **bool** f = **true**;  
 **for**(**int** j = 0; j < n; ++j){  
 **if** (B.A[i] == A[j]) f = **false**;  
 }  
 **if**(f) A[n++] = B.A[i];  
 }  
 A[n] = 0;  
 **return** \***this**;  
}  
Set Set::**operator** | (**const** Set & B) **const**{  
 Set C(\***this**);  
 **return** (C |= B);  
}  
  
Set & Set::**operator** = (**const** Set &B){  
 **if**(**this** != &B){  
 n = B.n;  
 S = 'R';  
 A = **new char** (B.n);  
 **int** i = 0;  
 **for**(;i < n; i++){  
 A[i] = B.A[i];  
 }  
 }  
 **return** \***this**;  
}  
  
**void** Set::show(){  
 std::cout << **this**->S << " =[" << A << "]" << std::endl;  
}

List.h:

#ifndef INC\_2\_LIST\_H  
#define INC\_2\_LIST\_H  
#include <iostream>  
#include <time.h>  
#endif //INC\_2\_LIST\_H  
  
**using namespace** std;  
  
**struct** list{  
 **char** a;  
 list \*next;  
 list(**char** b = '\0', list \*ne = **nullptr**){  
 a = b;  
 next = ne;  
 }  
 ~list(){  
 **delete** next;  
 }  
};  
  
**class** Set{  
**private**:  
 **static int** N;  
 **int** n;  
 **char** S;  
**public**:  
 list \*A = **nullptr**;  
 list \*A1 = A;  
 Set & **operator** | (Set&);  
 Set & **operator** |= (Set&);  
 Set & **operator** = (**const** Set &);  
 **void** show();  
 **int** power() {**return** n;};  
 Set(**char** S = '\0'){  
 cout << "Create " << S << endl;  
 **this**->S = S;  
 n = 0;  
 A = **nullptr**;  
 }  
 **void** generator(){  
 **int** wA = 0;  
 wA = rand() % 0xFFFF;  
 **for** (**int** i = 0; i < N - 1; i++){  
 **if** ((wA >> i) & 1){  
 A = **new** list((**char**)(i <= 9 ? i + '0' : i + 'A' - 10),A);  
 n++;  
 }  
 }  
 A1 = A;  
 cout << "generated: ";  
 **this**->show();  
 }  
 ~Set(){  
 cout << "delete: " << **this**->S << endl;  
 **delete** A;  
 }  
};  
  
**int** Set :: N = 16;  
  
Set & Set::**operator** |=(Set & B){  
 cout << "Operator |= got: ";  
 **this**->show();  
 cout << " & ";  
 B.show();  
 **while**(B.A != **nullptr**){  
 **bool** f = **true**;  
 **while**(A != **nullptr**){  
 **if** (B.A->a == A->a) f = **false**;  
 A = A->next;  
 }  
 **if**(f) {  
 A1 = **new** list (B.A->a,A1);  
 }  
 A = A1;  
 B.A = B.A->next;  
 }  
 B.A = B.A1;  
 cout << "return: ";  
 **this**->show();  
 **return** \***this**;  
}  
  
Set & Set::**operator** | (Set & B){  
 cout << "operator | got: ";  
 **this**->show();  
 cout << " & ";  
 B.show();  
 Set C(\***this**);  
 **return** (C |= B);  
}  
  
Set & Set::**operator** = (**const** Set &B){  
 cout << "operator = got: ";  
 **this**->show();  
 cout << " & ";  
 Set B11 = B;  
 B11.show();  
 cout << endl;  
  
 **if**(**this** != &B){  
 list \*B1 = B.A;  
 n = B.n;  
 S = 'R';  
 A = **nullptr**;  
 **while**(B1 != **nullptr**){  
 A = **new** list(B1->a,A);  
 B1 = B1->next;  
 }  
 }  
 **this**->A1 = **this**->A;  
 cout << "return: ";  
 **this**->show();  
 **return** \***this**;  
}  
  
**void** Set::show(){  
 cout << S << " = [";  
 **while** (A != **nullptr**){  
 cout << A->a;  
 A = A->next;  
 }  
 A = A1;  
 cout << "]" << endl;  
}

Bool.h:

#ifndef INC\_2\_BOOL\_H  
#define INC\_2\_BOOL\_H  
#include <iostream>  
#include <time.h>  
#endif //INC\_2\_BOOL\_H  
  
**using namespace** std;  
  
**class** Set{  
**private**:  
 **static int** N;  
 **int** n;  
 **char** S, \*A;  
**public**:  
 Set **operator** | (**const** Set&) **const**;  
 Set & **operator** |= (**const** Set&);  
 Set & **operator** = (**const** Set&);  
 **void** show();  
 **int** power() {**return** n;};  
 Set(**char** S1 = '\0'){  
 n = 0;  
 A = **nullptr**;  
 S = S1;  
 }  
 **void** generator(){  
 **int** wA = 0;  
 wA = rand() % 0xFFFF;  
 A = **new char** (N);  
 **for** (**int** i = 0; i < N - 1; i++){  
 **if** ((wA >> i) & 1){  
 A[n++] = '1';  
 } **else** A[n++] = '0';  
 }  
 }  
 ~Set(){**delete** [] A;}  
};  
  
**int** Set::N = 16;  
  
Set & Set::**operator** |= (**const** Set &B){  
 **for**(**int** i = 0; i < B.n; ++i){  
 A[i] = A[i] | B.A[i];  
 }  
 **return** \***this**;  
}  
  
Set Set::**operator** | (**const** Set &B) **const**{  
 Set C(\***this**);  
 **return** (C |= B);  
}  
  
Set & Set::**operator** = (**const** Set &B){  
 **if**(**this** != &B){  
 n = B.n;  
 S = 'R';  
 A = **new char** (B.n);  
 **int** i = 0;  
 **for**(;i < n; i++){  
 A[i] = B.A[i];  
 }  
 }  
 **return** \***this**;  
}  
  
**void** Set::show(){  
 std::cout << **this**->S << " =[" << A << "]" << std::endl;  
 std::cout << **this**->S << " =[";  
 **for** (**int** i = 0; i < N; i++){  
 **if** (A[i] == '1'){  
 cout << (**char**)(i <= 9 ? i + '0' : i + 'A' - 10);  
 }  
 }  
 cout << "]" << endl;  
}

Mw.h

#ifndef INC\_2\_BOOL\_H  
#define INC\_2\_BOOL\_H  
#include <iostream>  
#include <time.h>  
#endif //INC\_2\_BOOL\_H  
  
**using namespace** std;  
  
**class** Set{  
**private**:  
 **static int** N;  
 **int** n;  
 **char** S;  
 **short** A;  
**public**:  
 Set **operator** | (**const** Set&) **const**;  
 Set & **operator** |= (**const** Set&);  
 Set & **operator** = (**const** Set&);  
 **void** show();  
 **int** power() {**return** n;};  
 Set(**char** S1 = '\0'){  
 n = 0;  
 A = 0;  
 S = S1;  
 }  
 **void** generator(){  
 **int** wA = 0;  
 wA = rand() % 0xFFFF;  
 **for** (**int** i = 0; i < N - 1; i++){  
 **if** ((wA >> i) & 1){  
 A = wA | (1 << i);  
 n++;  
 }  
 }  
 }  
};  
  
**int** Set::N = 16;  
  
Set & Set::**operator** |= (**const** Set &B){  
 A |= B.A;  
 **return** \***this**;  
}  
  
Set Set::**operator** | (**const** Set &B) **const**{  
 Set C(\***this**);  
 **return** (C |= B);  
}  
  
Set & Set::**operator** = (**const** Set &B){  
 **if**(**this** != &B){  
 n = B.n;  
 S = 'R';  
 A = B.A;  
 }  
 **return** \***this**;  
}  
  
**void** Set::show(){  
 std::cout << **this**->S << " =[" << A << "]" << std::endl;  
 std::cout << **this**->S << " =[";  
 **for** (**int** i = 0; i < N; i++){  
 **if** ((A >> i) & 1) cout << (**char**)(i <= 9 ? i + '0' : i + 'A' - 10);  
 }  
 cout << "]" << endl;  
 cout << endl;  
 **for** (**int** i = 0; i < N; i++){  
 **if** ((A >> i) & 1) cout << 1; **else** cout << 0;  
 }  
 cout << endl;  
}