Лабораторная работа №7.

Текст программы для задания №6.

```
1.  #include <iostream>
2.  #include <string>
3.  #include <vector>
4.
5.  struct Abonent {
6.      unsigned int _number;
7.      std::string _secondName;
8.      unsigned long int _passNumber;
9.      Abonent() {}
10.     Abonent(unsigned int number, std::string secondName, unsigned long
11. int passNumber) {
12.         _number = number;
13.         _secondName = secondName;
14.         _passNumber = passNumber;
15.     }
16.     void input() {
17.         std::cout << "Input second name\n";
18.         std::cin >> _secondName;
19.         std::cout << "Input number\n";
20.         std::cin >> _number;
21.         std::cout << "Input number of passport\n";
22.         std::cin >> _passNumber;
23.     }
24.     void output() {
25.         std::cout << "=====================================\n";
26.         std::cout << "The abonent " << _secondName << '\n';
27.         std::cout << "Number: +" << _number << '\n';
28.         std::cout << "=====================================\n";
29.     }
30. };
31.
32. struct AbonentExtended : Abonent {
33.     long double _debtValue;
34.     AbonentExtended() {}
35.     AbonentExtended(unsigned int number, std::string secondName,
36. unsigned long int passNumber, long double debtValue) :
37.     Abonent(number, secondName, passNumber) {
```

```
38.         _debtValue = debtValue;
39.     }
40.     bool checkDebetor(long double base) {
41.         return !(_debtValue < base);
42.     }
43.     void input() {
44.         std::cout << "Input second name\n";
45.         std::cin >> _secondName;
46.         std::cout << "Input number\n";
47.         std::cin >> _number;
48.         std::cout << "Input number of passport\n";
49.         std::cin >> _passNumber;
50.         std::cout << "Input debt value\n";
51.         std::cin >> _debtValue;
52.     }
53.     void output() {
54.         std::cout << "=====================================\n";
55.         std::cout << "The abonent " << _secondName << '\n';
56.         std::cout << "Number: " << _number << '\n';
57.         std::cout << "Debt: " << _debtValue << '\n';
58.         std::cout << "=====================================\n";
59.     }
60. };
61.
62. int main() {
63.     unsigned int N;
64.     long double debtVal;
65.     std::cin >> N;
66.     std::vector<AbonentExtended> table =
67. std::vector<AbonentExtended>(N);
68.     for (auto it = table.begin(); it != table.end(); ++it) {
69.         it->input();
70.     }
71.     system("cls");
72.     std::cout << "Input debt base value: ";
73.     std::cin >> debtVal;
74.     std::cout << "Searching debetors...\n";
75.     for (auto it = table.begin(); it != table.end(); ++it) {
76.         if (it->checkDebetor(debtVal)) {
77.             it->output();
78.         }
79.     }
```

```
80.     return 0;
81. }
82.
```

Оценка характеристик программы

В тексте программы представлено 2 класса: Abonent и AbonentExtended. Рассчитаем вес каждого из классов:

$CS_{Abonent} = 3 + 2 = 5;$

$CS_{AbonentExtended} = 1 + 3 = 4;$

Теперь рассчитаем количество операций, переопределяемых классом (NOO):

$NOO_{AbonentExtended} = 2\ ;$

Далее рассчитаем значение NOA (количества операций, добавленных в класс):

$NOA_{AbonentExtended} = 1;$

Тогда Индекс специализации Si будет равен:

$$Si_{AbonentExtended} = \frac{NOO_{AbonentExtended} \cdot u}{M_{\text{общ}}} = \frac{2 \cdot 2}{3} = \frac{4}{3} \approx 1.33.$$

Теперь рассчитаем значение среднего размера операции AOS:

$AOS_{Abonent} = 0,$
поскольку не идёт прямое обращение к методам внешних классов

$AOS_{AbonentExtended} = 0,$
поскольку тоже не происходит обращение к методам внешних классов

$AOS_{main()} = 7,$
это я посчитал просто так, хотя $main(\quad)$ не является методом класса …

Далее рассмотрим сложность операций классов ОС на основе LOC-оценок:

Метод input класса Abonent:

Действие	Вес	Количество строк
Определение переменной-параметра	0.3	3
Определение временной переменной	0.5	0
Присваивание значения	0.5	0
Вложенное выражение	0.5	0
Сообщение без параметров	1	0
Арифм. операция	2	0
Сообщение с параметрами	3	0
Вызов стандартной функции интерфейса	5	6
Вызов пользовательской функции	7	0

$$OC_{Abonent::input()} = 0.9 + 30 = 30.9$$

Метод output класса Abonent:

Действие	Вес	Количество строк
Определение переменной-параметра	0.3	0
Определение временной переменной	0.5	0
Присваивание значения	0.5	0
Вложенное выражение	0.5	0
Сообщение без параметров	1	0
Арифм. операция	2	0
Сообщение с параметрами	3	0
Вызов стандартной функции интерфейса	5	4
Вызов пользовательской функции	7	0

$$OC_{Abonent::output()} = 20$$

Метод input класса AbonentExtended:

Действие	Вес	Количество строк
Определение переменной-параметра	0.3	4
Определение временной переменной	0.5	0
Присваивание значения	0.5	0
Вложенное выражение	0.5	0
Сообщение без параметров	1	0
Арифм. операция	2	0
Сообщение с параметрами	3	0
Вызов стандартной функции интерфейса	5	8
Вызов пользовательской функции	7	0

$$OC_{\text{AbonentExtended::}input()} = 1.2 + 40 = 41.2$$

Метод output класса Abonent:

Действие	Вес	Количество строк
Определение переменной-параметра	0.3	0
Определение временной переменной	0.5	0
Присваивание значения	0.5	0
Вложенное выражение	0.5	0
Сообщение без параметров	1	0
Арифм. операция	2	0
Сообщение с параметрами	3	0
Вызов стандартной функции интерфейса	5	5
Вызов пользовательской функции	7	0

$$OC_{\text{Abonent::}output()} = 25$$

Метод checkDebetor класса AbonentExtended:

Действие	Вес	Количество строк
Определение переменной-параметра	0.3	0
Определение временной переменной	0.5	0
Присваивание значения	0.5	0
Вложенное выражение	0.5	0
Сообщение без параметров	1	0
Арифм. операция	2	2
Сообщение с параметрами	3	0
Вызов стандартной функции интерфейса	5	0
Вызов пользовательской функции	7	0

$OC_{\text{AbonentExtended::checkDebetor()}} = 4;$

$ANP_{Abonent} = \frac{3}{2} = 1.5;$

$ANP_{AbonentExtended} = \frac{4}{3} = 1.33;$

Количество описаний сценариев:

$NSS_{Abonent} = 2,$

$NSS_{AbonentExtended} = 3,$

NKS = 1, т.к. оба класса в той или иной степени непосредственно контактируют с проблемной областью

$NSUB = 1, \text{т. к. } 1 \text{ подсистема} - \text{вывод должников}$