

Методические указания для СРС по выполнению лабораторных работ по дисциплине Объектно-ориентированное программирование

Тема: **Определение класса с динамически размещаемыми данными**

1. Реализовать класс с заданным интерфейсом на языке C++ согласно варианту.
2. Реализовать main с тестами для проверки класса (создание объекта и выполнение действий с ним)
3. Написать отчет
 - Постановка задачи
 - Описание интерфейса класса (class {} и комментарии ко всем полям, методам и функциям)
 - Описание тестов для проверки классов (main с комментариями, какие действия выполнялись, полученные результаты)
 - Листинг реализации класса (реализация методов и функций, отступы, без комментариев)

Шаблон отчета и пример заполнения отчета в приложенных файлах

4. Отправить отчет

Пример решения задания 1

Класс очередь целых чисел на циклическом буфере.

Операции:

конструктор(размер очереди), деструктор, очередьпуста?, очередьполна?,
добавить(значение), убратьпервый(), первый()

```
class Queue {  
    // поля  
    int *data; // массив для хранения элементов  
    int start, // первый элемент  
        n, // текущее количество  
        size; // максимальный размер  
public:  
    // методы  
    Queue(int); // конструктор  
    ~Queue() { delete[] data; } // деструктор  
    bool isEmpty() { return n==0; } // очередь пуста  
    bool isFull() { return n==size; } // очередь полна  
    void add(int x); // добавить  
    int first(); // первый  
    void del(); // удалить первый  
    void print(); // печать состояния очереди для проверки  
};  
Queue::Queue(int s) {  
    size=s;  
    n=0;  
    data=new int[s];  
    start=0;  
}
```

```

/* то же через список инициализации
Queue::Queue(int s): data(new int[s]), start(0), n(0), size(s) {
}
*/
void Queue::add(int x) {
    if(isFull()) {
        cout<<"очередь полна\n";
        exit(1);
    }
    data[(start+n)%size]=x;
    n++;
}
int Queue::first() {
    if(isEmpty()) {
        cout<<"очередь пуста\n";
        exit(1);
    }
    return data[start];
}
void Queue::del() {
    if(isEmpty()) return; // или ошибка
    start++;
    if(start==size) start=0;
    n--;
}
void Queue::print() {
    cout<<"индекс 1-го элемента "<<start<<" кол-во "<<n<<"\n";
    cout<<"элементы: ";
    for(int i=0;i<n;++i)
        cout<<data[(start+i)%size]<<" ";
    cout<<"\n";
}

```

Пример проверки (может быть короче), задача - вызвать все методы, время от времени печатая состояние объекта

```

int main() {
    Queue q(10);
    int x=1;
    while(!q.isFull()) {
        q.add(x++);
    }
    q.print();
    cout<<q.first()<<"\n";
    q.del();
    q.add(100);
    q.print();
    while(!q.isEmpty()) {
        cout<<q.first()<<"\n";
        q.del();
    }
    q.print();
}

```

Варианты заданий

Вариант 1

Сортированный набор строк

```
class SList
{
public:
    SList(int maxsize=100);
    ~SList();
    void add(char *str);    // добавить строку
    int size();            // количество строк
    int find(char *str);    // найти строку, вернуть -1 если нет
    void remove(int);    // удалить строку
    char *item(int);    // строка по номеру
    void print();    // распечатать состояние объекта
};
```

Вариант 2

Множество целых от а до b Использовать массив флагов, i-й флаг указывает наличие числа

```
i+a
class ASet
{
public:
    ASet(int a, int b);
    ASet(int b);    // а по умолчанию 0
    ~ASet();
    void incl(int);    // включить число в множество
                        // если такое число уже есть, то ничего не делать
    void excl(int);    // исключить
    bool contain(int);    // содержится в множестве?
    void print();    // распечатать состояние объекта
};
```

Остальные варианты <https://ipc.susu.ru/20768-2.html>

Южно-Уральский государственный университет (НИУ)
Институт естественных и точных наук
Кафедра «Прикладная математика и программирование»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1
по дисциплине «Объектно-ориентированное программирование»

Автор работы
студент группы ЕТ-212
_____ А.А.Александрова
_____ 2017 г.

Работа зачтена с оценкой

_____ А.К.Демидов
_____ 2017 г.

Челябинск, 2017

1 Постановка задачи

I. Реализовать класс

1. Ассоциативная таблица для представления списков вида *имя=значение* и словарей *fish*
рыба

```
class ATable
{
    public:
        ATable(int maxsize=100);
        ~ATable();
        void add(char *name, char *value);
        void remove(char *name);
        void setValue(char *name, char *value); //
заменить //значение
        char *getValue(char *name);           // если не нашли
//вернуть NULL
        void print(); // распечатать состояние объекта
};
```

II. Реализовать main с тестами для проверки класса (создание объекта и выполнение действий с ним)

2 Описание интерфейса класса

```
class ATable {
    int size, // текущий размер словаря
        maxsize; // максимальный размер словаря.
    struct word {
        char *x; // слово
        char *y; // перевод слова
    } *a; // массив ассоциаций
public:
    ATable(int maxsize=100):size(0),maxsize(maxsize), a(new abc[maxsize])
    {} // конструктор
    ~ATable() { delete [] a; } // деструктор
    void add(char *name, char *value); // добавление нового слова в словарь
    void remove(char *name); // удаляет слово
    void setValue(char *name, char *value); // заменить значение
    char *getValue(char *name); // если не нашли вернуть NULL
    void print(); // распечатать состояние объекта
};
```

3 Описание тестов для проверки классов

```
int main() {  
    ATable A;  
    cout<<"1. Создаем список из 3 слов\n";  
    A.add("fish", "рыба");//добавляем слова  
    A.add("cat", "кот");  
    A.add("dog", "собака");  
    A.print();//выводит на экран содержание словаря  
    cout<<"2. Удалим слово cat\n";  
    A.remove("cat");//удаляет слово  
    A.print();  
    cout<<"3. Замена значения у 1-го слова\n";  
    A.setValue("fish", "рыбачить");//заменяет перевод слова  
    A.print();  
    cout<<"4. Вывод значения 1-го слова\n";  
    cout<<A.getValue("fish")<<endl;//выводит перевод слова  
    return 0;  
}
```

Полученные результаты

1. Введем список из 3 слов

fish рыба

cat кот

dog собака

2. Удалим слово cat

fish рыба

dog собака

3. Замена значения у 1-го слова

fish рыбачить

dog собака

4. Вывод значения 1-го слова

собака

4 Листинг реализации класса

```
void ATable:: add(char *name, char *value) {  
    if(size==maxsize){  
        cout<<"Таблица заполнена\n";  
        exit(1);  
    }  
    strcpy(a[size].x=new char[strlen(name)+1],name);  
    strcpy(a[size].y= new char[strlen(value)+1],value);  
    ++size;  
};
```