# Лекция 11

# 1. Разработка методов полиморфного класса DBTableSet

//-----чтение всех таблиц БД из файлов в db-------------------

int DBTableSet::ReadDB()

{//чтение имен таблиц БД из файла DBTables.txt в strArray

db.clear();

string path="..\\"+dbName+"\\";

vector<string> strArray;//буфер для чтения таблицы DBTables.txt

ifstream fin(path+"DBTables.txt");

if (!fin.is\_open())

{

cout<<"Ошибка открытия файла\n"<< endl;

system("pause");

return -1;

}

const int len=200;//максимальная длина строки таблицы

char line[len];

char \*token = NULL;

char \*next\_token = line;//NULL;

fin.getline(line, len);//в таблице DBTables.txt всего 1 строка с именами таблиц БД

while(token = strtok\_s( next\_token, "|",&next\_token))

strArray.push\_back(token);//для нового элемента память выделяет vector

//(в классе vector перегружен оператор присваивания -> "глубокое" копирование)

string endOfdbName=dbName.substr(dbName.size()-3,3);//БД текстовая или бинарная?

for(unsigned int i=0; i<strArray.size(); i++)

{ //реализация полиморфизма:

strcpy\_s(line,len,strArray[i].c\_str());

line[strlen(line)-4]='\0';//убрать .txt или .bin из имени таблицы

if(endOfdbName=="Bin")

db[line]=new DBTableBin(line);//бинарная таблица с именем line if(endOfdbName=="Txt")

db[line]=new DBTableTxt(line);//текстовая таблица с именем line db[line]->ReadDBTable(path+strArray[i]);//чтение в таблицу данных из файла

// ReadDBTable() – виртуальная функция, тип db[line] – DBTable\*. В зависимости от //типа БД (txt или bin), будет вызываться либо DBTableTxt::ReadDBTable(),   
// либо DBTableBin::ReadDBTable().

}

return 0;

}

# 2. Техническая документация на программные продукты