Таблица UML

|  |  |
| --- | --- |
| << class >>  pSTRING | Output |
| -string\* pStr; |  |
| << Конструкторы >>  +pSTRING();  +pSTRING(string str);  +pSTRING(const pSTRING& obj);  +~pSTRING();  << Методы >>  +void TO\_SET(string str);  +string TO\_GET\_VAL();  +string& TO\_GET\_VAR();  +char& FRONT();  +char& BACK();  +void SWAP(pSTRING& obj);  +void PUSH\_BACK(char ch);  +void TO\_SHOW();  +int LENGTH();  +void ASSIGN(const pSTRING& obj);  +pSTRING operator = (const pSTRING& obj); |  |
| Класс для хранения в куче информации об одной строке и работы с ней. |  |

Реализация методов

|  |
| --- |
| // Конструкторы  pSTRING::pSTRING() { pStr = new string(""); }  pSTRING::pSTRING(string str) { pStr = new string(str); }  pSTRING::pSTRING(const pSTRING& obj) { this->pStr = new string(\*(obj.pStr)); }  pSTRING::~pSTRING() { delete pStr; }  // Методы  void pSTRING::TO\_SET(string str) { \*pStr = str; }  string pSTRING::TO\_GET\_VAL() { return \*pStr; }  string& pSTRING::TO\_GET\_VAR() { return \*pStr; }  char& pSTRING::FRONT() { return (\*pStr)[0]; }  char& pSTRING::BACK() { return (\*pStr)[pStr->size() - 1]; }  void pSTRING::SWAP(pSTRING& obj) {  string temp = TO\_GET\_VAL();  TO\_SET(obj.TO\_GET\_VAL());  obj.TO\_SET(temp);  }  void pSTRING::PUSH\_BACK(char ch) { \*pStr += ch; }  void pSTRING::TO\_SHOW() { cout << \*pStr << endl; }  int pSTRING::LENGTH() { return pStr->size(); }  void pSTRING::ASSIGN(const pSTRING& obj) { delete pStr; this->pStr = new string(\*obj.pStr); }  pSTRING pSTRING::operator = (const pSTRING& obj) { ASSIGN(obj); return \*this; } |

Main()

|  |
| --- |
| pSTRING a, b("hello"), c(b);  b.TO\_SHOW(); c.TO\_SHOW(); c.TO\_SET("goodbye"); c.TO\_SHOW(); b = c;b.TO\_SHOW();  cout << b.FRONT() << ' ' << b.BACK() << endl;  b.PUSH\_BACK('1'); b.TO\_SHOW(); b.SWAP(c); b.TO\_SHOW(); c.TO\_SHOW(); |