

Балтийский государственный технический университет
«ВОЕНМЕХ» им. Д. Ф. Устинова

Кафедра О7 «Информационные системы и программная инженерия»

Практическая работа №1
по дисциплине «Структуры и организация данных»
на тему «Линейные структуры данных»

вариант 20

Выполнил:
Студент Кабиров А.Р.
Группа О713Б

Преподаватель:
Кимсанбаев К.А..

Санкт-Петербург
2022 г

Дополнительные методы для класса «Строка-идентификатор»: поиск первого вхождения символа в строку, переопределение операции меньше

Производный от «Строки» класс Шестнадцатеричная строка

Строки данного класса могут содержать только символы шестнадцатеричных цифр (как в верхнем, так и в нижнем регистре) и символы - и +, задающие знак числа, которые могут находиться только в первой позиции числа, при отсутствии знака число считается положительным. Если в составе инициализирующей строки будут встречены любые символы, отличные от допустимых, Шестнадцатеричная строка принимает нулевое значение. Содержимое данных строк рассматривается как шестнадцатеричное число.

Обязательные методы: перевод всех буквенных символов строки в верхний регистр, сложения +, вычитания -

Главный класс String:

```
class String {
public:
    char *m_string;

    // дефолтный конструктор
    String();

    // конструктор принимающий на вход строку
    String(char *string);

    // конструктор копирования
    String(const String &str);

    // деструктор
    ~String();

    // перегрузка оператора '='
    String &operator=(String &otherString);

};
```

Унаследованный от класса String класс IdentifiedString:

```
class IdentifiedString : public String {
public:
    // дефолтный конструктор
    IdentifiedString() : String() {
        m_string = new char[100];
    }

    // конструктор принимающий в качестве параметра строку
    IdentifiedString(char *string) : String(string) {
    // проверка введенной строки на наличие пробелов.
    // если есть пробелы то строка будет пустой
        setString(string);
    }

    // конструктор копирования
    IdentifiedString(const IdentifiedString &str);

    // деструктор
    ~IdentifiedString();

    // проверка строки, если содержит пробелы записать пустую строку
    void setString(char *string);
};
```

```
//    получение первого вхождения символа в строку
int IndexOf(char c);

};
```

Унаследованный от класса IdentifiedString класс HexString:

```
class HexString : public IdentifiedString {
private:
//    переменная hexNumbers показывает какие символы..
//    ..может содержать строка
    const char *hexNumbers = "+-0123456789AaBbCcDdEeFf";
//    длина числа (знак не считается).
//    удобно для конвертации из hex в dec и в обратную
    int length;

//    метод переводит символы в десятичные числа
    int char2dec(char hexChar);

public:
//    обозначает какого знака число
    bool isPositive = true;

//    наследуемы конструктор
    HexString() : IdentifiedString() { m_string = new char[100]; };

//    деструктор
    ~HexString();

//    конструктор принимающий в качестве параметра строку
    HexString(char *string) : IdentifiedString(string) {
        setString(string);
    }

//    конструктор копирования
    HexString(const HexString &str);

//    Сеттер поля m_string
    void setString(char *string);

//    перевод буквенной части числа в верх. регистр
    void toUpper();

//    перевод буквенной части числа в нижн. регистр
    void toLower();

//    перевод hex строки в десятичное число
    int hexToDec();

//    десятичное число в hex строку
    char *decToHex(int n);

//    перегрузка оператора +
    HexString *operator+(HexString *anotherHex);

//    перегрузка оператора -
    HexString *operator-(HexString *anotherHex);

};
```

Основной код программы:

```
int main() {
    char s[] = "11aDf";
    HexString hex1 = HexString(s);
    HexString hex2 = HexString("-A");
    HexString *hex3 = hex1 + &hex2;

    cout << "Как написано: " << hex1.m_string << endl;
    hex1.toLower();
    cout << "В нижнем регистре: " << hex1.m_string << endl;
    hex1.toUpper();
    cout << "В верхнем регистре: " << hex1.m_string << endl;

    cout << "Индекс 'a': " << hex1.IndexOf('a') << endl;

    cout << hex1.m_string << " + " << hex2.m_string << " = " << hex3-
>m_string << endl;

    hex3 = hex1 - &hex2;
    cout << hex1.m_string << " - " << hex2.m_string << " = " << hex3-
>m_string << endl;

    HexString hexEmpty = HexString("234laf d");
    cout << "Пустая строка потому что есть недопустимые знаки или пробел" <<
hexEmpty.m_string << endl;
    return 0;
}
```

```
using namespace std;
```

```
int main() {
    char s[] = "11aDf";
    HexString hex1 = HexString(s);
    HexString hex2 = HexString( string: "-A");
    HexString *hex3 = hex1 + &hex2;

    cout << "Как написано: " << hex1.m_string << endl;
    hex1.toLower();
    cout << "В нижнем регистре: " << hex1.m_string << endl;
    hex1.toUpper();
    cout << "В верхнем регистре: " << hex1.m_string << endl;

    cout << "Индекс 'a': " << hex1.IndexOf( c: 'a') << endl;

    cout << hex1.m_string << " + " << hex2.m_string << " = " << hex3->m_string << endl;

    hex3 = hex1 - &hex2;
    cout << hex1.m_string << " - " << hex2.m_string << " = " << hex3->m_string << endl;

    HexString hexEmpty = HexString( string: "234laf d");
    cout << "Пустая строка потому что есть недопустимые знаки или пробел" << hexEmpty.m_string << endl;
    return 0;
}
```

Как написано: 11aDf

В нижнем регистре: 11adf

В верхнем регистре: 11ADF

Индекс 'A': 2

11ADF + -A = 1DA15

11ADF - -A = 1EA19

Пустая строка потому что есть недопустимые знаки или пробел

Destructor HexString

Destructor IdentifiedString