## 1 . 문제정의

* 문자열을 분석하여 문자열에서 알파벳 문자의 빈도를 분석하여 필요시 출력하는 ` Histogram ` 클래스를 구현하라.
* 문자열은 대소문자 구분없이 소문자로 분류되어야 한다.

## 2 . 해결과정

\* 자료구조

- 히스토그램은 각각의 알파벳이 얼마나 많이 있는지를 저장해야 하므로, 각 알파벳의 빈도를 저장하는 배열인 int[] : chart를 둔다. (index는 각 알파벳의 순서와 같다.)

- 알파벳만의 길이인 int : length를 둔다.

- 분석 대상인 문자열인 string : text를 둔다.

\* 전제.

- 문자열들은 모두 char의 집합이므로, 문자열을 한 문자마다 분리하여 처리하는 분할 정복을 시행한다.

\* 메소드 함수 정의

## 생성자 (base : string)

* 생성자는 문자열 하나를 받아 실행한다
* Chart를 0으로 초기화한다. (하지 않는다면 쓰레기 값이 나올 우려가 있다.)
* 그 다음, 자신에게 put(base)을 실행하여 문자열에 대한 처리를 한다.

## put (s : string)

* Put는 분석대상 문자열을 삽입하는 함수이다.
* String은 char들로 이루어져 있으므로, 문자열 하나하나를 putc(s[n])로 삽입하여 통계처리를 한다.

## putc (c : char)

* Putc는 분석대상 문자 하나를 삽입하는 함수이다.
* 만약 분석대상 문자를 소문자로 변환 시도하고, 그 변환된 문자가 알파벳이라면 c-‘a’를 통해 몇 번째 알파벳인지 찾는다. (가능하다면 Hash table을 써도 좋긴 하다)
* 또한, 위에서 알파벳이 맞다면, length를 하나 더 늘린다.
* 위에서 구한 알파벳 인덱스로 chart에 접근하여, 그 값을 하나 증가시킨다.
* 이후, 원본 문자열 (self.text)에 c를 추가하여 원본문자열을 저한다.

## print

- text를 출력한다.

- 26글자를 모두 순회하며 chart의 원소 값만큼 for문을 돌아 \*을 출력한다.

# MAIN의 구성

- 객체 생성 후 put/putc로 추가한다.

- 이후 완료되면 delete를 통하여 불요한 전역메모리를 반환한다.

## 3. 소스코드

|  |
| --- |
| /\*\*  \* @file 06주과제.cpp  \* @author 2017011 이준영 (presan100@gmail.com)  \* @brief c++ 06주 과제  \* @version 0.1  \* @date 2024-04-08  \*  \* @copyright Copyright (c) 2024  \*  \*/  #include <iostream>  #include <string>  using namespace std;  class Histogram  {  private :  int chart[26]; // 문자들의 통계  int length = 0; // 알파벳 문자열 길이  string text; // 원본  public :  Histogram(string base)  {  for(int q = 0; q < 26 ; q ++)  {  // 배열 내부 초가화  chart[q] = 0;  }  // 처음 들어온 문자열 초기화  this->put(base);  }  ~Histogram()  {  }  void put(string s)  {  for(int q = 0; q<s.length() ; q++){  // char[]이므로 char로 분할정복한다.  this->putc(s[q]);  }  }  void putc(char c)  {  char lower = tolower(c);// 소문자화.  if (lower >= 'a' && lower <= 'z')  {  // 만약 알파벳일때.  this->length ++; // 전체 알파벳수 증가  int idx = (lower - 'a');  chart[idx] = chart[idx] + 1; // 부분원소 증가  }  text += c; // 원본 추가.  }  void print(){  cout << this->text << endl;  cout << "총 알파벳 수" << this->length << endl;  for (int code = 0 ; code < 26 ; code ++){  // 알파벳을 26개 이므로 26개에 걸쳐 순회  char now = code + 'a';  int c = this->chart[code];  cout << now << '[' << c << ']' << ':';  for (int selector = 0 ; selector < c ; selector ++ )  {  cout << '\*'; // 별표기  }  cout << endl;  }  }  };  int main()  {  Histogram\* a = new Histogram("zwhat is this?:>");  a->put("HAHA HHHHHHAAAAAA");  a->put("Den, der doppelt elend ist,");  a->putc('-');  a->put("I would just copy a book or something");  a->putc('c');  a->print();  delete a; // 메모리 반납  } |

4. 실행결과.

텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명