Аленов Михаил Константинович БПИ204 2 курс

Архитектура Вычислительных Систем – Дз 3 Вариант 121

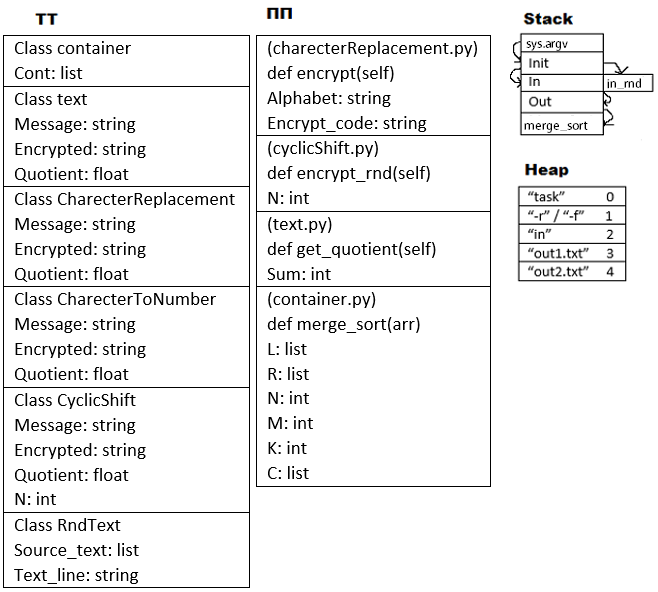
**Задание:**

|  |  |  |  |
| --- | --- | --- | --- |
| Обобщенный артефакт, используемый в задании | Базовые альтернативы (уникальные параметры, задающие отличительные признаки альтернатив) | Общие для всех альтернатив переменные | Общие для всех альтернатив функции |
| 9. Тексты, состоящие из цифр и латинских букв, зашифрованные различными способами. | 1. Шифрование заменой символов (указатель на массив пар: [текущий символ, замещающий символ]; зашифрованный текст – строка символов)  2. Шифрование циклическим сдвигом кода каждого символа на n (целое число, определяющее сдвиг; зашифрованный текст – строка символов)  3. Шифрование заменой символов на числа (пары: текущий символ, целое число – подстановка при шифровании кода символа в виде короткого целого; | Открытый текст – строка символов. | Частное от деления суммы кодов незашифрованной строки на число символов в этой строке (действительное число) |

**Функция:**

9. Упорядочить элементы контейнера по возрастанию используя сортировку с помощью прямого слияния (Straight Merge). В качестве ключей для сортировки и других действий используются результаты функции, общей для всех альтернатив.

**Структурная схема программы:**



**Основные характеристики программы:**

**Число интерфейсных модулей** – 0

**Число модулей реализации** – 7

**Общий размер исходных текстов** – 14.1 КБ

**Используемая память и время работы программы для тестовых прогонов**:

Тесты расположены в подкаталоге:

**Testing\in-data\**

Результаты работы тестовых прогонов расположены в подкаталоге:

**Testing\out-data\**

Неслучайные осмысленные тесты:

|  |  |  |
| --- | --- | --- |
| Ввод | Память (кб) | Время (микросекунды) |
| python3 main.py -f Testing\in-data\test1.txt Testing\out-data\test1-out1.txt Testing\out-data\test1-out2.txt | 4688 | 1173 |
| python3 main.py -f Testing\in-data\test2.txt Testing\out-data\test2-out1.txt Testing\out-data\test2-out2.txt | 4596 | 2209 |
| python3 main.py -f Testing\in-data\test3.txt Testing\out-data\test3-out1.txt Testing\out-data\test3-out2.txt | 4572 | 4730 |
| python3 main.py -f Testing\in-data\test4.txt Testing\out-data\test4-out1.txt Testing\out-data\test4-out2.txt | 4588 | 2239 |
| python3 main.py -f Testing\in-data\test5.txt Testing\out-data\test5-out1.txt Testing\out-data\test5-out2.txt | 4624 | 1874 |

Случайные неосмысленные тесты:

|  |  |  |
| --- | --- | --- |
| python3 main.py -r 5 Testing\out-data\rand1-out1.txt Testing\out-data\rand1-out2.txt | 4632 | 2021 |
| python3 main.py -r 100 Testing\out-data\rand2-out1.txt Testing\out-data\rand2-out2.txt | 4776 | 8067 |
| Python3 main.py -r 1000 Testing\out-data\rand3-out1.txt Testing\out-data\rand3-out2.txt | 5316 | 65958 |
| Python3 main.py -r 10000 Testing\out-data\rand4-out1.txt Testing\out-data\rand4-out2.txt | 10912 | 638716 |

Столбец памяти является интерпретацией результатов работы программы в Windows Task Manager

Столбец времени является интерпретацией результатов работы программы, полученных при помощи python библиотеки time

**Сравнения с характеристиками ранее разработанных программ:**

**Обще для ДЗ-1 и ДЗ-2:**

Уменьшился вес исходных файлов программы, так как в ДЗ-1 и ДЗ-2 использовались файлы заголовков и реализации.

Используемая программой память значительно возросла, что связано с динамической типизацией python.

Скорость работы программы ниже, чем в ДЗ-1 и ДЗ-2, из-за того, что python – высокоуровневый интерпретируемый ЯП, в отличие от компилируемый ЯП, таких как С и С++

**Проверка корректности работы программы на примере прогона файла test1.txt**

**Ввод в терминал:** python3 main.py -f Testing\in-data\test1.txt Testing\out-data\test1-out1.txt Testing\out-data\test1-out2.txt

|  |  |  |
| --- | --- | --- |
| Содержимое test1.txt: | Содержимое test1-out1.txt: | Содержимое test1-out2.txt: |
| 1  Object  2  2 Object  3  Object | Filled container:  Container contains 3 elements  0)  Text: Object  Encryption by substitution of characters: Gwptez  1)  Text: Object  Encryption by cyclically shifting the code of each character by 2: (n = 2) Qdlgev  2)  Text: Object  Encryption by replacing characters with numbers: 79 98 106 101 99 116 | Sorted container  Container contains 3 elements  0)  Quotient = 99.83333333333333  Text: Object  Encryption by replacing characters with numbers: 79 98 106 101 99 116  1)  Quotient = 99.83333333333333  Text: Object  Encryption by cyclically shifting the code of each character by 2: (n = 2) Qdlgev  2)  Quotient = 99.83333333333333  Text: Object  Encryption by substitution of characters: Gwptez |

Посчитаем частное для слова Object:

O = 79, b = 98, j = 106, e = 101, c = 99, t = 116

Сумма кодов = 599

Частное = 599 / 6 = 99.83

При шифровке слова Object при помощи:

1. Замены символов, получается слово Gwptez
2. Циклического сдвига на 2, получается слово Qdlgev
3. Замены символа на его численный эквивалент в ASCII, получается слово 79 98 106 101 99 116

Расшифруем полученные слова:

1. Используем алфавит qwertyuiopasdfghjklzxcvbnm.

G -> O

w -> b

p -> j

t -> e

e -> c

z -> t

1. Сдвину все символы в слове Qdlgev на 2 позиции влево

Q -> O

d -> b

I -> j

g -> e

e -> c

v -> t

1. Переведу все символы в буквы, используя <https://commons.wikimedia.org/wiki/File:ASCII-Table.svg>

79 -> O

98 -> b

106 -> j

101 -> e

99  -> c

116 -> t

Получается, программа корректно шифрует сообщения и находит частное.