Лаб. работа №4. Шифр Гронсвельда.

Создано системой Doxygen 1.9.4

2.1 Файлы 3 3 Классы 5 3.1 Класс cipher_error 5 3.1.1 Подробное описание 5 3.1.2 Конструктор(ы) 5 3.1.2.1 cipher_error() [1/2] 5 3.1.2.2 cipher_error() [2/2] 6 3.1.3 Meroды 6 3.1.3 u what() 6 3.1.4 Данные класса 6 3.1.4.1 msg 6 3.2 Класс modAlphaCipher 7 3.2.1 Подробное описание 7 3.2.2 Конструктор(ы) 7 3.2.2 конструктор(ы) 7 3.2.2 modAlphaCipher() [1/2] 8 3.2.3 методы 9 3.2.3.1 convert() [1/2] 9 3.2.3.2 convert() [2/2] 9 3.2.3.3 decrypt() 9 3.2.3.4 encrypt() 10 3.2.3.5 getValidCipherText() 10 3.2.3.7 getValidOpenText() 11 3.2.4.1 alphaNum 11 3.2.4.2 key 11 3.2.4.3 numAlpha 11	1 Алфавитный указатель классов	1
2.1 Φαλπы 3 3 Κлассы 5 3.1. Класс cipher_error 5 3.1.2 Конструктор(ы) 5 3.1.2.1 cipher_error() [1/2] 5 3.1.2.2 cipher_error() [2/2] 6 3.1.3.1 what() 6 3.1.4 Данные класса 6 3.1.4 цвв 7 3.2 Класс modAlphaCipher 7 3.2.1 подробное описание 7 3.2.2 конструктор(ы) 7 3.2.2.1 modAlphaCipher() [1/2] 8 3.2.2.2 modAlphaCipher() [1/2] 8 3.2.3.3 методы 9 3.2.3.4 сонует() [1/2] 9 3.2.3.2 getValidCipherText() 10 3.2.3.3 getValidCipherText() 10 3.2.3.7 getValidOpenText() 11 3.2.4.1 alphaNum 11 3.2.4.2 key 11 3.2.4.3 numAlpha 11	1.1 Классы	. 1
3 Κπαεσεί 5 3.1 Κπαες cipher_error 5 3.1.1 Ποπροδιώς ο απικαιμε 5 3.1.2 Κοικτργκτορ(ω) 5 3.1.2.1 cipher_error() [1/2] 5 3.1.2.2 cipher_error() [2/2] 6 3.1.3 Merogus 6 3.1.4 Данные класса 6 3.1.4 Данные класса 6 3.2 Класс modAlphaCipher 7 3.2.1 Подробное описание 7 3.2.2 Конструктор(ы) 7 3.2.2.1 modAlphaCipher() [1/2] 8 3.2.2.2 modAlphaCipher() [2/2] 8 3.2.3 Merogus 9 3.2.3.3 convert() [1/2] 9 3.2.3.3 decrypt() 9 3.2.3.5 getValidCipherText() 10 3.2.3.7 getValidKey() 10 3.2.4 данные класса 11 3.2.4.1 аlphaNum 11 3.2.4.2 key 11 3.2.4.3 numAlpha 11 4 Файлы 11 4.1 Файл тайн.срр 13 4.1.1 Файл тайн.срр 13 4.1.1 Фикини 13 4.1.1.1 i Svalid() 14 <td>2 Список файлов</td> <td>3</td>	2 Список файлов	3
3.1. Класс cipher_error 5 3.1.1 Подробное описание 5 3.1.2 Конструктор(ы) 5 3.1.2.1 cipher_error() [1/2] 5 3.1.2.2 cipher_error() [2/2] 6 3.1.3 Merogus 6 3.1.4 Данные класса 6 3.1.4.1 msg 6 3.2 Класс modAlphaCipher 7 3.2.1 Подробное описание 7 3.2.2 Конструктор(ы) 7 3.2.2.1 modAlphaCipher() [1/2] 8 3.2.2.2 modAlphaCipher() [2/2] 8 3.2.3.1 convert() [1/2] 9 3.2.3.2 convert() [2/2] 9 3.2.3.3 decrypt() 9 3.2.3.4 encrypt() 10 3.2.3.5 getValidCipherText() 10 3.2.3.6 getValidCipherText() 10 3.2.3.7 getValidOpenText() 11 3.2.4.1 alphaNum 11 3.2.4.2 key 11 3.2.4.3 numAlpha 11 4 Файлы 13 4.1 Файлы 13 4.1.1 Фуикции 13 4.1.1 файл main.cpp 13 4.1.1.1 isValid() 14	2.1 Файлы	. 3
3.1.1 Подробное описание 5 3.1.2 Конструктор(ы) 5 3.1.2.1 cipher_error() [1/2] 5 3.1.2.2 cipher_error() [2/2] 6 3.1.3 Meroды 6 3.1.3.1 what() 6 3.1.4 Данные класса 6 3.1.4.1 msg 6 3.2 Класс modAlphaCipher 7 3.2.1 Подробное описание 7 3.2.2 Конструктор(ы) 7 3.2.2.1 modAlphaCipher() [1/2] 8 3.2.2.2 modAlphaCipher() [2/2] 8 3.2.3.1 convert() [1/2] 9 3.2.3.2 convert() [2/2] 9 3.2.3.3 decrypt() 9 3.2.3.5 getValidCipherText() 10 3.2.3.5 getValidCopenText() 10 3.2.3.7 getValidOpenText() 11 3.2.4.1 alphaNum 11 3.2.4.2 key 11 3.2.4.3 numAlpha 11 4 Файлы 13 4.1 Файлынин 13 4.1.1 Функции 13 4.1.1 Функции 13 4.1.1.1 isValid() 14	3 Классы	5
3.1.2 Конструктор(ы) 5 3.1.2.1 cipher_error() [1/2] 5 3.1.2.2 cipher_error() [2/2] 6 3.1.3 Meroды 6 3.1.3.1 what() 6 3.1.4 Данные класса 6 3.2.4 Класс modAlphaCipher 7 3.2.1 Подробное описание 7 3.2.2 Конструктор(ы) 7 3.2.2.1 modAlphaCipher() [1/2] 8 3.2.2.2 modAlphaCipher() [2/2] 8 3.2.3 Meroды 9 3.2.3.1 convert() [1/2] 9 3.2.3.2 convert() [2/2] 9 3.2.3.3 decrypt() 9 3.2.3.4 encrypt() 10 3.2.3.5 getValidCipherText() 10 3.2.3.7 getValidOpenText() 11 3.2.4.1 alphaNum 11 3.2.4.2 key 11 3.2.4.3 numAlpha 11 4 Файлы 13 4.1 Файлы 13 4.1 Файлы 13 4.1.1 Функции 13 4.1.1 функции 13 4.1.1.1 isValid() 14	3.1 Kласс cipher_error	. 5
3.1.2.1 cipher_error() [1/2] 5 3.1.2.2 cipher_error() [2/2] 6 3.1.3 Metoqus 6 3.1.3.1 what() 6 3.1.4 Данные класса 6 3.1.4.1 msg 6 3.2 Класс modAlphaCipher 7 3.2.1 Подробное описание 7 3.2.2 Конструктор(ы) 7 3.2.2.1 modAlphaCipher() [1/2] 8 3.2.3.2.2 modAlphaCipher() [2/2] 8 3.2.3.1 convert() [1/2] 9 3.2.3.2 convert() [2/2] 9 3.2.3.3 decrypt() 9 3.2.3.4 encrypt() 10 3.2.3.5 getValidCipherText() 10 3.2.3.7 getValidOpenText() 11 3.2.4.1 alphaNum 11 3.2.4.2 key 11 3.2.4.3 numAlpha 11 4 Файлы 13 4.1 Файлы маіл.срр 13 4.1.1 Функция 13 4.1.1 Функция 13 4.1.1.1 isValid() 14	3.1.1 Подробное описание	. 5
3.1.2.2 cipher_error() [2/2] 6 3.1.3 Методы 6 3.1.3.1 what() 6 3.1.4 Данные класса 6 3.1.4.1 msg 6 3.2.1 Подробное описание 7 3.2.2 Конструктор(ы) 7 3.2.2.1 modAlphaCipher() [1/2] 8 3.2.2.2 modAlphaCipher() [2/2] 8 3.2.3 Meroды 9 3.2.3.2 convert() [1/2] 9 3.2.3.3 decrypt() 9 3.2.3.4 encrypt() 10 3.2.3.5 getValidCipherText() 10 3.2.3.6 getValidKey() 10 3.2.3.7 getValidOpenText() 11 3.2.4.1 alphaNum 11 3.2.4.2 key 11 3.2.4.3 numAlpha 11 4 Файлы 13 4.1 Файлы 13 4.1.1 функции 13 4.1.1 функции 13 4.1.1 функции 13 4.1.1 isValid() 14	3.1.2 Конструктор(ы)	. 5
3.1.3 Методы 6 3.1.4 Данные класса 6 3.1.4.1 msg 6 3.2.1 Подробное описание 7 3.2.2 Конструктор(ы) 7 3.2.2.1 modAlphaCipher() [1/2] 8 3.2.2.2 modAlphaCipher() [2/2] 8 3.2.3 Методы 9 3.2.3.1 convert() [1/2] 9 3.2.3.2 convert() [2/2] 9 3.2.3.3 decrypt() 9 3.2.3.4 encrypt() 10 3.2.3.5 getValidCipherText() 10 3.2.3.6 getValidKey() 10 3.2.3.7 getValidOpenText() 11 3.2.4.1 alphaNum 11 3.2.4.2 key 11 3.2.4.3 numAlpha 11 4 Файлы 13 4.1 Файлы 13 4.1.1 функции 13 4.1.1 функции 13 4.1.1 іsValid() 14	$3.1.2.1 \; \mathrm{cipher_error}() \; [1/2] \; \ldots \; $. 5
3.1.3.1 what() 6 3.1.4 Данные класса 6 3.1.4.1 msg 6 3.2 Класс modAlphaCipher 7 3.2.1 Подробное описание 7 3.2.2 Конструктор(ы) 7 3.2.2.1 modAlphaCipher() [1/2] 8 3.2.2.2 modAlphaCipher() [2/2] 8 3.2.3.3 Meroды 9 3.2.3.1 convert() [1/2] 9 3.2.3.2 convert() [2/2] 9 3.2.3.3 decrypt() 9 3.2.3.4 encrypt() 10 3.2.3.5 getValidCipherText() 10 3.2.3.7 getValidOpenText() 11 3.2.4 Данные класса 11 3.2.4.1 alphaNum 11 3.2.4.2 key 11 3.2.4.3 numAlpha 11 4 Файлы 11 4.1 Файлы майлысрр 13 4.1.1 Функции 13 4.1.1.1 isValid() 14	$3.1.2.2 ext{ cipher_error}() [2/2] \dots \dots \dots \dots \dots \dots \dots \dots \dots$. 6
3.1.4 Данные класса 6 3.1.4.1 msg 6 3.2 Класс modAlphaCipher 7 3.2.1 Подробное описание 7 3.2.2 Конструктор(ы) 7 3.2.2.1 modAlphaCipher() [1/2] 8 3.2.2.2 modAlphaCipher() [2/2] 8 3.2.3.3 Meroды 9 3.2.3.1 convert() [1/2] 9 3.2.3.2 convert() [2/2] 9 3.2.3.3 decrypt() 9 3.2.3.4 encrypt() 10 3.2.3.5 getValidCipherText() 10 3.2.3.7 getValidOpenText() 11 3.2.4 Данные класса 11 3.2.4.1 alphaNum 11 3.2.4.2 key 11 3.2.4.3 numAlpha 11 4 Файлы 11 4.1 Файлы майлысрр 13 4.1.1 Функции 13 4.1.1.1 isValid() 14	3.1.3 Методы	. 6
3.1.4.1 msg 6 3.2 Класс modAlphaCipher 7 3.2.1 Подробное описание 7 3.2.2 Конструктор(ы) 7 3.2.2.1 modAlphaCipher() [1/2] 8 3.2.2.2 modAlphaCipher() [2/2] 8 3.2.3 Metoqia 9 3.2.3.1 convert() [1/2] 9 3.2.3.2 convert() [2/2] 9 3.2.3.3 decrypt() 9 3.2.3.4 encrypt() 10 3.2.3.5 getValidCipherText() 10 3.2.3.6 getValidKey() 10 3.2.3.7 getValidOpenText() 11 3.2.4 Данные класса 11 3.2.4.1 alphaNum 11 3.2.4.2 key 11 3.2.4.3 numAlpha 11 4 Файлы 13 4.1 Файл таіл.срр 13 4.1.1 Функции 13 4.1.1.1 isValid() 14	$3.1.3.1 \text{ what}() \dots \dots$. 6
3.2 Класс modAlphaCipher 7 3.2.1 Подробное описание 7 3.2.2 Конструктор(ы) 7 3.2.2.1 modAlphaCipher() [1/2] 8 3.2.2.2 modAlphaCipher() [2/2] 8 3.2.3 Meroды 9 3.2.3.1 convert() [1/2] 9 3.2.3.2 convert() [2/2] 9 3.2.3.3 decrypt() 9 3.2.3.4 encrypt() 10 3.2.3.5 getValidCipherText() 10 3.2.3.6 getValidKey() 10 3.2.3.7 getValidOpenText() 11 3.2.4 Данные класса 11 3.2.4.1 аlphaNum 11 3.2.4.2 key 11 3.2.4.3 numAlpha 11 4 Файлы 13 4.1 Файл тып.срр 13 4.1.1 Функции 13 4.1.1.1 isValid() 14	3.1.4 Данные класса	. 6
3.2.1 Подробное описание 7 3.2.2 Конструктор(ы) 7 3.2.2.1 modAlphaCipher() [1/2] 8 3.2.2.2 modAlphaCipher() [2/2] 8 3.2.3 Meroды 9 3.2.3.1 convert() [1/2] 9 3.2.3.2 convert() [2/2] 9 3.2.3.3 decrypt() 9 3.2.3.4 encrypt() 10 3.2.3.5 getValidCipherText() 10 3.2.3.6 getValidKey() 10 3.2.3.7 getValidOpenText() 11 3.2.4.1 alphaNum 11 3.2.4.2 key 11 3.2.4.3 numAlpha 11 4 Файлы 13 4.1 Файл таin.cpp 13 4.1.1 Функции 13 4.1.1 Функции 13 4.1.1.1 isValid() 14	3.1.4.1 msg	. 6
3.2.2 Конструктор(ы) 7 3.2.2.1 modAlphaCipher() [1/2] 8 3.2.2.2 modAlphaCipher() [2/2] 8 3.2.3 Meroды 9 3.2.3.1 convert() [1/2] 9 3.2.3.2 convert() [2/2] 9 3.2.3.3 decrypt() 9 3.2.3.4 encrypt() 10 3.2.3.5 getValidCipherText() 10 3.2.3.6 getValidKey() 10 3.2.3.7 getValidOpenText() 11 3.2.4 Данные класса 11 3.2.4.1 alphaNum 11 3.2.4.2 key 11 3.2.4.3 numAlpha 11 4 Файлы 13 4.1 Файл таin.cpp 13 4.1.1 Функции 13 4.1.1.1 isValid() 14	3.2 Kлаcc modAlphaCipher	. 7
3.2.2.1 modAlphaCipher() [1/2] 8 3.2.2.2 modAlphaCipher() [2/2] 8 3.2.3 Meтоды 9 3.2.3.1 convert() [1/2] 9 3.2.3.2 convert() [2/2] 9 3.2.3.3 decrypt() 9 3.2.3.4 encrypt() 10 3.2.3.5 getValidCipherText() 10 3.2.3.6 getValidKey() 10 3.2.3.7 getValidOpenText() 11 3.2.4 Данные класса 11 3.2.4.1 alphaNum 11 3.2.4.2 key 11 3.2.4.3 numAlpha 11 4 Файлы 13 4.1.1 Функции 13 4.1.1 Функции 13 4.1.1.1 isValid() 14	3.2.1 Подробное описание	. 7
3.2.2.1 modAlphaCipher() [1/2] 8 3.2.2.2 modAlphaCipher() [2/2] 8 3.2.3 Meтоды 9 3.2.3.1 convert() [1/2] 9 3.2.3.2 convert() [2/2] 9 3.2.3.3 decrypt() 9 3.2.3.4 encrypt() 10 3.2.3.5 getValidCipherText() 10 3.2.3.6 getValidKey() 10 3.2.3.7 getValidOpenText() 11 3.2.4 Данные класса 11 3.2.4.1 alphaNum 11 3.2.4.2 key 11 3.2.4.3 numAlpha 11 4 Файлы 13 4.1.1 Функции 13 4.1.1 Функции 13 4.1.1.1 isValid() 14		
3.2.2.2 modAlphaCipher() [2/2] 8 3.2.3 Методы 9 3.2.3.1 convert() [1/2] 9 3.2.3.2 convert() [2/2] 9 3.2.3.3 decrypt() 9 3.2.3.4 encrypt() 10 3.2.3.5 getValidCipherText() 10 3.2.3.6 getValidKey() 10 3.2.3.7 getValidOpenText() 11 3.2.4 Данные класса 11 3.2.4.1 alphaNum 11 3.2.4.2 key 11 3.2.4.3 numAlpha 11 4 Файлы 13 4.1 Файл таin.cpp 13 4.1.1 Функции 13 4.1.1.1 isValid() 14	3.2.2.1 modAlphaCipher() [1/2]	. 8
3.2.3 Методы93.2.3.1 convert() [1/2]93.2.3.2 convert() [2/2]93.2.3.3 decrypt()93.2.3.4 encrypt()103.2.3.5 getValidCipherText()103.2.3.6 getValidKey()103.2.3.7 getValidOpenText()113.2.4 Данные класса113.2.4.1 alphaNum113.2.4.2 key113.2.4.3 numAlpha114 Файлы134.1 Файл таin.cpp134.1.1 Функции134.1.1 функции134.1.1.1 isValid()14		
3.2.3.1 convert() [1/2] 9 3.2.3.2 convert() [2/2] 9 3.2.3.3 decrypt() 9 3.2.3.4 encrypt() 10 3.2.3.5 getValidCipherText() 10 3.2.3.6 getValidKey() 10 3.2.3.7 getValidOpenText() 11 3.2.4 Данные класса 11 3.2.4.1 alphaNum 11 3.2.4.2 key 11 3.2.4.3 numAlpha 11 4 Файлы 13 4.1.1 Файл таіп.срр 13 4.1.1 Функций 13 4.1.1.1 isValid() 14	- · · · · · · · · · · · · · · · · · · ·	
3.2.3.2 convert() [2/2] 9 3.2.3.3 decrypt() 9 3.2.3.4 encrypt() 10 3.2.3.5 getValidCipherText() 10 3.2.3.6 getValidKey() 10 3.2.3.7 getValidOpenText() 11 3.2.4 Данные класса 11 3.2.4.1 alphaNum 11 3.2.4.2 key 11 3.2.4.3 numAlpha 11 4 Файлы 13 4.1 Файл таіп.срр 13 4.1.1 Функции 13 4.1.1.1 isValid() 14		
3.2.3.3 decrypt() 9 3.2.3.4 encrypt() 10 3.2.3.5 getValidCipherText() 10 3.2.3.6 getValidKey() 10 3.2.3.7 getValidOpenText() 11 3.2.4 Данные класса 11 3.2.4.1 alphaNum 11 3.2.4.2 key 11 3.2.4.3 numAlpha 11 4 Файлы 13 4.1 Файл таin.cpp 13 4.1.1 Функции 13 4.1.1.1 isValid() 14	Washing	
3.2.3.4 encrypt() 10 3.2.3.5 getValidCipherText() 10 3.2.3.6 getValidKey() 10 3.2.3.7 getValidOpenText() 11 3.2.4 Данные класса 11 3.2.4.1 alphaNum 11 3.2.4.2 key 11 3.2.4.3 numAlpha 11 4 Файлы 13 4.1 Файл main.cpp 13 4.1.1 Функции 13 4.1.1.1 isValid() 14		
3.2.3.5 getValidCipherText() 10 3.2.3.6 getValidKey() 10 3.2.3.7 getValidOpenText() 11 3.2.4 Данные класса 11 3.2.4.1 alphaNum 11 3.2.4.2 key 11 3.2.4.3 numAlpha 11 4 Файлы 13 4.1 Файл таin.cpp 13 4.1.1 Функции 13 4.1.1.1 isValid() 14	V = V	
3.2.3.6 getValidKey() 10 3.2.3.7 getValidOpenText() 11 3.2.4 Данные класса 11 3.2.4.1 alphaNum 11 3.2.4.2 key 11 3.2.4.3 numAlpha 11 4 Файлы 13 4.1 Файл таin.cpp 13 4.1.1 Функции 13 4.1.1.1 isValid() 14		
3.2.3.7 getValidOpenText() 11 3.2.4 Данные класса 11 3.2.4.1 alphaNum 11 3.2.4.2 key 11 3.2.4.3 numAlpha 11 4 Файлы 13 4.1 Файл main.cpp 13 4.1.1 Функции 13 4.1.1.1 isValid() 14	- "	
3.2.4 Данные класса 11 3.2.4.1 alphaNum 11 3.2.4.2 key 11 3.2.4.3 numAlpha 11 4 Файлы 13 4.1 Файл main.cpp 13 4.1.1 Функции 13 4.1.1.1 isValid() 14		
3.2.4.1 alphaNum 11 3.2.4.2 key 11 3.2.4.3 numAlpha 11 4 Файлы 13 4.1 Файл main.cpp 13 4.1.1 Функции 13 4.1.1.1 isValid() 14		
3.2.4.2 key 11 3.2.4.3 numAlpha 11 4 Файлы 13 4.1 Файл main.cpp 13 4.1.1 Функции 13 4.1.1.1 isValid() 14		
3.2.4.3 numAlpha 11 4 Файлы 13 4.1 Файл main.cpp 13 4.1.1 Функции 13 4.1.1.1 isValid() 14		
4.1 Файл main.cpp 13 4.1.1 Функции 13 4.1.1.1 isValid() 14	•	
4.1 Файл main.cpp 13 4.1.1 Функции 13 4.1.1.1 isValid() 14	4 Файлы	13
4.1.1 Функции 13 4.1.1.1 isValid() 14	4.1 Файл main.cpp	
4.1.1.1 isValid()		
· ·	· ·	
11112 11111()	v ·	
4.2 Файл modAlphaCipher.cpp	· ·	
4.3 Файл modAlphaCipher.h		
4.3.1 Подробное описание		
4.4 modAlphaCipher.h		
4.5 Файл modException.h		

4.5.1 Подробное описание	17
4.6 modException.h	17
Предметный указатель	19

Алфавитный указатель классов

1.1 Классы

Классы с их кратким описанием.

cipher_error	
Класс для ошибок	5
modAlphaCipher	
Класс для зашифрования и расшифрования текста	7

Алфавитный	указатель	классов
TITOMORITIDIA	Y IXAOA I CAID	12/10/00/1

Список файлов

2.1 Файлы

Полный список файлов.

main.cpp	13
modAlphaCipher.cpp	14
modAlphaCipher.h	
Файл с описание класса modAlphaCipher	4
modException.h	
Файл с описанием класса ошибок	16

4 Список файлов

Классы

3.1 Класс cipher_error

Класс для ошибок

#include <modException.h>

Открытые члены

- cipher_error (const std::wstring &what_arg) Конструктор со строкой.
- cipher_error (const wchar_t *what_arg) Конструктор с массивом символов.
- std::wstring what () const Получить сообщение.

Закрытые данные

• std::wstring msg Сообщение об ошибке.

3.1.1 Подробное описание

Класс для ошибок

3.1.2 Конструктор(ы)

Конструктор со строкой.

Аргументы

in what_arg Сообщение.	in
------------------------	----

```
3.1.2.2 cipher_error() [2/2]
```

Конструктор с массивом символов.

Аргументы

in what_ar	g Сообщение.
------------	--------------

3.1.3 Методы

```
3.1.3.1 \text{ what}()
```

```
std::wstring\ cipher\_error::what\ (\ )\ const\quad [inline]
```

Получить сообщение.

Возвращает

std::wstring Сообщение.

3.1.4 Данные класса

```
3.1.4.1 \text{ msg}
```

 $std::wstring\ cipher_error::msg\quad [private]$

Сообщение об ошибке.

Объявления и описания членов класса находятся в файле:

 $\bullet \ \, \mathrm{modException.h}$

3.2 Класс modAlphaCipher

Класс для зашифрования и расшифрования текста.

#include <modAlphaCipher.h>

Открытые члены

- modAlphaCipher ()=delete
- modAlphaCipher (const std::wstring &skey)

Конструктор с ключом.

• std::wstring encrypt (const std::wstring &open text)

Зашифровать текст.

• std::wstring decrypt (const std::wstring &cipher text)

Расшифровать текст.

Закрытые члены

• std::vector< int > convert (const std::wstring &s)

Преобразовать строку в вектор.

• std::wstring convert (const std::vector< int > &v)

Преобразовать вектор в строку.

• std::wstring getValidKey (const std::wstring &s)

Проверить и получить ключ.

• std::wstring getValidOpenText (const std::wstring &s)

Проверить и получить открытый текст.

• std::wstring getValidCipherText (const std::wstring &s)

Проверить и получить шифротекст.

Закрытые данные

- std::wstring numAlpha = L"АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ" алфавит по порядку
- std::map< char, int > alphaNum

ассоциативный массив "номер по символу"

• std::vector< int > key

3.2.1 Подробное описание

Класс для зашифрования и расшифрования текста.

3.2.2 Конструктор(ы)

$3.2.2.1 \mod Alpha Cipher()$ [1/2]

 ${\bf modAlphaCipher::} {\bf modAlphaCipher} \; (\;) \quad [{\bf delete}]$

$3.2.2.2 \mod Alpha Cipher() [2/2]$

 ${\bf modAlphaCipher::modAlphaCipher\ (}$ ${\bf const\ std::wstring\ \&\ skey\)}$

Конструктор с ключом.

Аргументы

```
in skey Ключ.
```

3.2.3 Методы

```
3.2.3.1 convert() [1/2]
```

```
std::wstring mod
Alpha<br/>Cipher::convert ( {\rm const~std::vector} < {\rm int} > \& {\rm ~v~}) \quad [{\rm inline}], [{\rm private}]
```

Преобразовать вектор в строку.

Аргументы

in v Вектор инден	ссов.	
-------------------	-------	--

Возвращает

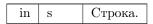
std::vector<int> Строка.

3.2.3.2 convert() [2/2]

```
std::vector < int > modAlphaCipher::convert \; ( \\ const \; std::wstring \; \& \; s \; ) \quad [inline], \; [private]
```

Преобразовать строку в вектор.

Аргументы



Возвращает

std::vector<int> Вектор индексов.

```
3.2.3.3 decrypt()
```

```
std::wstring modAlphaCipher::decrypt ( const std::wstring & cipher_text )
```

Расшифровать текст.

Аргументы

in cipher_text	Зашифрованный текст
----------------	---------------------

Возвращает

std::wstring Открытый текст.

```
3.2.3.4 encrypt()
```

```
std::wstring modAlphaCipher::encrypt ( {\rm const~std::wstring~\&~open\_text~)}
```

Зашифровать текст.

Аргументы

in open_text	Открытый текст.
--------------	-----------------

Возвращает

std::wstring Зашифрованный текст.

3.2.3.5 getValidCipherText()

```
std::wstring modAlphaCipher::getValidCipherText ( {\rm const~std::wstring~\&~s~)} \quad [{\rm private}]
```

Проверить и получить шифротекст.

Аргументы

in	s	Шифротекст.

Возвращает

std::wstring Обработанная строка.

3.2.3.6 getValidKey()

```
std::wstring modAlphaCipher::getValidKey ( {\rm const~std::wstring~\&~s~)} \quad [{\rm inline}], \, [{\rm private}]
```

Проверить и получить ключ.

Аргументы

in s	Строка-ключ.
------	--------------

Возвращает

std::wstring Обработанный ключ.

3.2.3.7 getValidOpenText()

```
std::wstring\ modAlphaCipher::getValidOpenText\ ( const\ std::wstring\ \&\ s\ )\quad [private]
```

Проверить и получить открытый текст.

Аргументы

in	s	Открытый текст.
----	---	-----------------

Возвращает

std::wstring Обработанная строка.

3.2.4 Данные класса

3.2.4.1 alphaNum

```
std::map<char,int> modAlphaCipher::alphaNum [private] ассоциативный массив "номер по символу"
```

3.2.4.2 key

std::vector<int> modAlphaCipher::key [private]

3.2.4.3 numAlpha

std::wstring modAlphaCipher::numAlpha = L"АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЦЪЫЬЭЮЯ" [private] алфавит по порядку

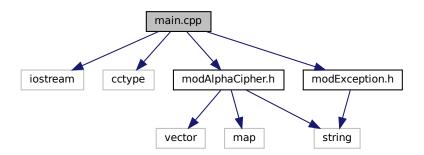
Объявления и описания членов классов находятся в файлах:

- modAlphaCipher.h
- $\bullet \ \operatorname{modAlphaCipher.cpp}$

Файлы

4.1 Файл таіп.срр

```
#include <iostream>
#include <cctype>
#include "modAlphaCipher.h"
#include "modException.h"
Граф включаемых заголовочных файлов для main.cpp:
```



Функции

- bool is Valid (const wstring &s)
- int main (int argc, char **argv)

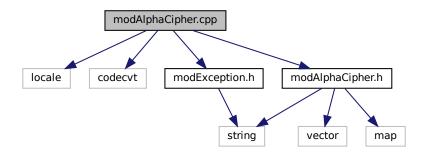
4.1.1 Функции

14 Файлы

4.1.1.1 isValid()

4.2 Файл modAlphaCipher.cpp

```
#include <locale>
#include <codecvt>
#include "modException.h"
#include "modAlphaCipher.h"
Граф включаемых заголовочных файлов для modAlphaCipher.cpp:
```



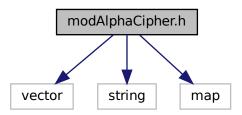
4.3 Файл modAlphaCipher.h

Файл с описание класса modAlphaCipher.

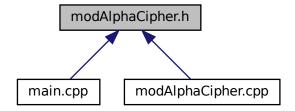
```
#include <vector>
#include <string>
```

#include <map>

Граф включаемых заголовочных файлов для modAlphaCipher.h:



Граф файлов, в которые включается этот файл:



Классы

 \bullet class modAlphaCipher

Класс для зашифрования и расшифрования текста.

4.3.1 Подробное описание

Файл с описание класса modAlphaCipher.

Автор

Шурыгин Д.Д.

Дата

2023-12-25

16 Файлы

4.4 modAlphaCipher.h

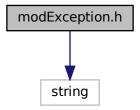
```
См. документацию.
\begin{array}{c} 1 \\ 7 \ \# \mathrm{pragma} \ \mathrm{once} \end{array}
9 #include <vector>
10 #include <string>
11 #include <map>
12
16 class modAlphaCipher {
17 private:
             std::wstring numAlpha = L"АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ";
18
19
              std::map <char,int> alphaNum;
              std::vector <int> key; // ключ
std::vector<int> convert(const std::wstring& s);
20
27
34
             std::wstring convert(const std::vector<int>& v);
std::wstring getValidKey(const std::wstring & s);
std::wstring getValidOpenText(const std::wstring & s);
41
48
              std::wstring getValidCipherText(const std::wstring & s);
57
              {\bf modAlphaCipher()}{=}{\bf delete};
63
70
77
78 };
             modAlphaCipher(const std::wstring& skey);
std::wstring encrypt(const std::wstring& open_text);
std::wstring decrypt(const std::wstring& cipher_text);
```

4.5 Файл modException.h

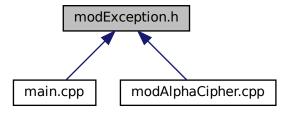
Файл с описанием класса ошибок

#include <string>

Граф включаемых заголовочных файлов для modException.h:



Граф файлов, в которые включается этот файл:



4.6 modException.h

Классы

```
    class cipher_error
    Класс для ошибок
```

4.5.1 Подробное описание

Файл с описанием класса ошибок

Автор

Шурыгин Д.Д.

Дата

2023-12-25

4.6 modException.h

```
Cm. документацию.
1
7 #include <string>
8
12 class cipher_error {
13 private:
14 std::wstring msg;
15 public:
21 cipher_error(const std::wstring& what_arg): msg(what_arg) {
37 explicit cipher_error(const wchar_t* what_arg): msg(what_arg) {
38 std::wstring what()const { return msg; }
39 std::wstring what()const { return msg; }
```

18 Файлы

Предметный указатель

```
alphaNum
                                                      numAlpha
    modAlphaCipher, 11
                                                          modAlphaCipher, 11
{\rm cipher\_error},\, {\color{red} 5}
                                                      what
    cipher error, 5, 6
                                                           cipher error, 6
    msg, 6
    what, 6
convert
    modAlphaCipher, 9
decrypt
    modAlphaCipher, 9
encrypt
    modAlphaCipher, 10
getValidCipherText
    modAlphaCipher, 10
{\rm getValidKey}
    \bmod Alpha Cipher,\, {\color{red}10}
getValidOpenText\\
    modAlphaCipher, 11
isValid
    main.cpp, 13
key
    modAlphaCipher, 11
main
    main.cpp, 14
main.cpp, 13
    isValid, 13
    main, 14
modAlphaCipher, 7
    alphaNum, 11
    convert, 9
    decrypt, 9
    encrypt, 10
    getValidCipherText, 10
    getValidKey, 10
    getValidOpenText, 11
    key, 11
    modAlphaCipher, 7, 8
    numAlpha, 11
{\it modAlphaCipher.cpp,}~{\it 14}
modAlphaCipher.h, 14
modException.h, 16
msg
```

cipher error, 6