

Algorytmy i programowanie - programowanie obiektowe - klasa string

Marcin Żurowski

10 maja 2021

string - konstruktory

- `string s1;`
- `string s2("tekst");`
- `string s3("tekst", 3);`
- `string s3(5, '*');`
- `string s3(s2, 2, 2);`

string - konstruktory

- `string s1;`
- `string s2("tekst");`
- `string s3("tekst", 3);`
- `string s3(5, '*');`
- `string s3(s2, 2, 2);`

string - konstruktory

- `string s1;`
- `string s2("tekst");`
- `string s3("tekst", 3);`
- `string s3(5, '*');`
- `string s3(s2, 2, 2);`

string - konstruktory

- `string s1;`
- `string s2("tekst");`
- `string s3("tekst", 3);`
- `string s3(5, '*');`
- `string s3(s2, 2, 2);`

string - konstruktory

- `string s1;`
- `string s2("tekst");`
- `string s3("tekst", 3);`
- `string s3(5, '*');`
- `string s3(s2, 2, 2);`

string - operator

● =

● +

● +=

● []

string - operator

● =

● +

● +=

● []

string - operator

- =
- +
- +=
- []

string - operator

- =
- +
- +=
- []

string - metody

- `size_type size();`
- `size_type length();`
- `bool empty();`
- `size_type max_size();`
- `size_type capacity();`
- `void reserve(size_type ile);`
- `void resize(size_type ile, char znak = '\\0');`
- `void clear();`
- `char & at(size_type pozycja);`
- `string substr(size_type pozycja, size_type ile_znakow);`

string - metody

- `size_type size();`
- `size_type length();`
- `bool empty();`
- `size_type max_size();`
- `size_type capacity();`
- `void reserve(size_type ile);`
- `void resize(size_type ile, char znak = '\\0');`
- `void clear();`
- `char & at(size_type pozycja);`
- `string substr(size_type pozycja, size_type ile_znakow);`

string - metody

- `size_type size();`
- `size_type length();`
- `bool empty();`
- `size_type max_size();`
- `size_type capacity();`
- `void reserve(size_type ile);`
- `void resize(size_type ile, char znak = '\\0');`
- `void clear();`
- `char & at(size_type pozycja);`
- `string substr(size_type pozycja, size_type ile_znakow);`

string - metody

- `size_type size();`
- `size_type length();`
- `bool empty();`
- `size_type max_size();`
- `size_type capacity();`
- `void reserve(size_type ile);`
- `void resize(size_type ile, char znak = '\\0');`
- `void clear();`
- `char & at(size_type pozycja);`
- `string substr(size_type pozycja, size_type ile_znakow);`

string - metody

- `size_type size();`
- `size_type length();`
- `bool empty();`
- `size_type max_size();`
- `size_type capacity();`
- `void reserve(size_type ile);`
- `void resize(size_type ile, char znak = '\\0');`
- `void clear();`
- `char & at(size_type pozycja);`
- `string substr(size_type pozycja, size_type ile_znakow);`

string - metody

- `size_type size();`
- `size_type length();`
- `bool empty();`
- `size_type max_size();`
- `size_type capacity();`
- `void reserve(size_type ile);`
- `void resize(size_type ile, char znak = '\\0');`
- `void clear();`
- `char & at(size_type pozycja);`
- `string substr(size_type pozycja, size_type ile_znakow);`

string - metody

- `size_type size();`
- `size_type length();`
- `bool empty();`
- `size_type max_size();`
- `size_type capacity();`
- `void reserve(size_type ile);`
- `void resize(size_type ile, char znak = '\\0');`
- `void clear();`
- `char & at(size_type pozycja);`
- `string substr(size_type pozycja, size_type ile_znakow);`

string - metody

- `size_type size();`
- `size_type length();`
- `bool empty();`
- `size_type max_size();`
- `size_type capacity();`
- `void reserve(size_type ile);`
- `void resize(size_type ile, char znak = '\\0');`
- `void clear();`
- `char & at(size_type pozycja);`
- `string substr(size_type pozycja, size_type ile_znakow);`

string - metody

- `size_type size();`
- `size_type length();`
- `bool empty();`
- `size_type max_size();`
- `size_type capacity();`
- `void reserve(size_type ile);`
- `void resize(size_type ile, char znak = '\\0');`
- `void clear();`
- `char & at(size_type pozycja);`
- `string substr(size_type pozycja, size_type ile_znakow);`

string - metody

- `size_type size();`
- `size_type length();`
- `bool empty();`
- `size_type max_size();`
- `size_type capacity();`
- `void reserve(size_type ile);`
- `void resize(size_type ile, char znak = '\\0');`
- `void clear();`
- `char & at(size_type pozycja);`
- `string substr(size_type pozycja, size_type ile_znakow);`

string - metody - find

- `size_type find(const char * zadany_c_string, size_type odkąd_zacząć = 0);`
- `size_type find(const string & zadany_string, size_type odkąd_zacząć = 0);`
- `size_type find(const char * wzór, size_type odkąd_zacząć, size_type długość);`
- `size_type find(char znak, size_type odkąd_zacząć = 0);`

string - metody - find

- `size_type find(const char * zadany_c_string, size_type odkąd_zacząć = 0);`
- `size_type find(const string & zadany_string, size_type odkąd_zacząć = 0);`
- `size_type find(const char * wzór, size_type odkąd_zacząć, size_type długość);`
- `size_type find(char znak, size_type odkąd_zacząć = 0);`

string - metody - find

- `size_type find(const char * zadany_c_string, size_type odkąd_zacząć = 0);`
- `size_type find(const string & zadany_string, size_type odkąd_zacząć = 0);`
- `size_type find(const char * wzór, size_type odkąd_zacząć, size_type długość);`
- `size_type find(char znak, size_type odkąd_zacząć = 0);`

string - metody - find

- `size_type find(const char * zadany_c_string, size_type odkąd_zacząć = 0);`
- `size_type find(const string & zadany_string, size_type odkąd_zacząć = 0);`
- `size_type find(const char * wzór, size_type odkąd_zacząć, size_type długość);`
- `size_type find(char znak, size_type odkąd_zacząć = 0);`

string - metody - rfind

- `size_type rfind(const char * zadany_c_string, size_type odkąd_zacząć = npos);`
- `size_type rfind(const string & zadany_string, size_type odkąd_zacząć = npos);`
- `size_type rfind(const char * wzór, size_type odkąd_zacząć, size_type długość);`
- `size_type rfind(char znak, size_type odkąd_zacząć = npos);`

string - metody - rfind

- `size_type rfind(const char * zadany_c_string, size_type odkąd_zacząć = npos);`
- `size_type rfind(const string & zadany_string, size_type odkąd_zacząć = npos);`
- `size_type rfind(const char * wzór, size_type odkąd_zacząć, size_type długość);`
- `size_type rfind(char znak, size_type odkąd_zacząć = npos);`

string - metody - rfind

- `size_type rfind(const char * zadany_c_string, size_type odkąd_zacząć = npos);`
- `size_type rfind(const string & zadany_string, size_type odkąd_zacząć = npos);`
- `size_type rfind(const char * wzór, size_type odkąd_zacząć, size_type długość);`
- `size_type rfind(char znak, size_type odkąd_zacząć = npos);`

string - metody - rfind

- `size_type rfind(const char * zadany_c_string, size_type odkąd_zacząć = npos);`
- `size_type rfind(const string & zadany_string, size_type odkąd_zacząć = npos);`
- `size_type rfind(const char * wzór, size_type odkąd_zacząć, size_type długość);`
- `size_type rfind(char znak, size_type odkąd_zacząć = npos);`

string - metody - find_of

- find_first_of
- find_last_of
- find_first_not_of
- find_last_not_of

string - metody - find_of

- find_first_of
- find_last_of
- find_first_not_of
- find_last_not_of

string - metody - find_of

- find_first_of
- find_last_of
- find_first_not_of
- find_last_not_of

string - metody - find_of

- find_first_of
- find_last_of
- find_first_not_of
- find_last_not_of

string - metody - usuwanie i wstawianie

- `string& erase(size_type pozycja = 0, size_type ile_znaków = npos);`
- `string& insert(size_type gdzie_wstawić, const string & co_wstawić);`
- `string& insert(size_type gdzie_wstawić, const string & co_wstawić, size_type skąd_zacząć, size_type ile_znaków);`
- `string& insert(size_type gdzie_wstawić, const char * co_wstawić);`
- `string& insert(size_type gdzie_wstawić, const char * co_wstawić, size_type ile_znaków);`
- `string& insert(size_type gdzie_wstawić, size_type ile_znaków, char znak);`

string - metody - usuwanie i wstawianie

- `string& erase(size_type pozycja = 0, size_type ile_znaków = npos);`
- `string& insert(size_type gdzie_wstawić, const string & co_wstawić);`
- `string& insert(size_type gdzie_wstawić, const string & co_wstawić, size_type skąd_zacząć, size_type ile_znaków);`
- `string& insert(size_type gdzie_wstawić, const char * co_wstawić);`
- `string& insert(size_type gdzie_wstawić, const char * co_wstawić, size_type ile_znaków);`
- `string& insert(size_type gdzie_wstawić, size_type ile_znaków, char znak);`

string - metody - usuwanie i wstawianie

- `string& erase(size_type pozycja = 0, size_type ile_znaków = npos);`
- `string& insert(size_type gdzie_wstawić, const string & co_wstawić);`
- `string& insert(size_type gdzie_wstawić, const string & co_wstawić, size_type skąd_zacząć, size_type ile_znaków);`
- `string& insert(size_type gdzie_wstawić, const char * co_wstawić);`
- `string& insert(size_type gdzie_wstawić, const char * co_wstawić, size_type ile_znaków);`
- `string& insert(size_type gdzie_wstawić, size_type ile_znaków, char znak);`

string - metody - usuwanie i wstawianie

- `string& erase(size_type pozycja = 0, size_type ile_znaków = npos);`
- `string& insert(size_type gdzie_wstawić, const string & co_wstawić);`
- `string& insert(size_type gdzie_wstawić, const string & co_wstawić, size_type skąd_zacząć, size_type ile_znaków);`
- `string& insert(size_type gdzie_wstawić, const char * co_wstawić);`
- `string& insert(size_type gdzie_wstawić, const char * co_wstawić, size_type ile_znaków);`
- `string& insert(size_type gdzie_wstawić, size_type ile_znaków, char znak);`

string - metody - usuwanie i wstawianie

- `string& erase(size_type pozycja = 0, size_type ile_znaków = npos);`
- `string& insert(size_type gdzie_wstawić, const string & co_wstawić);`
- `string& insert(size_type gdzie_wstawić, const string & co_wstawić, size_type skąd_zacząć, size_type ile_znaków);`
- `string& insert(size_type gdzie_wstawić, const char * co_wstawić);`
- `string& insert(size_type gdzie_wstawić, const char * co_wstawić, size_type ile_znaków);`
- `string& insert(size_type gdzie_wstawić, size_type ile_znaków, char znak);`

string - metody - usuwanie i wstawianie

- `string& erase(size_type pozycja = 0, size_type ile_znaków = npos);`
- `string& insert(size_type gdzie_wstawić, const string & co_wstawić);`
- `string& insert(size_type gdzie_wstawić, const string & co_wstawić, size_type skąd_zacząć, size_type ile_znaków);`
- `string& insert(size_type gdzie_wstawić, const char * co_wstawić);`
- `string& insert(size_type gdzie_wstawić, const char * co_wstawić, size_type ile_znaków);`
- `string& insert(size_type gdzie_wstawić, size_type ile_znaków, char znak);`

string - metody - zamiana

- `string& replace(size_type odkąd_zastępować, size_type ile_zastąpić, const string & co_wstawić);`
- `string& replace(size_type odkąd_zastępować, size_type ile_zastąpić, const string & co_wstawić, size_type odkąd_brać, size_type ile_znaków);`
- `string& replace(size_type odkąd_zastępować, size_type ile_zastąpić, const char * co_wstawić, size_type ile_znaków);`
- `string& replace(size_type odkąd_zastępować, size_type ile_zastąpić, const char * co_wstawić);`
- `string& replace(size_type odkąd_zastępować, size_type ile_zastąpić, size_type ile_znaków, char znak);`

string - metody - zamiana

- `string& replace(size_type odkąd_zastępować, size_type ile_zastąpić, const string & co_wstawić);`
- `string& replace(size_type odkąd_zastępować, size_type ile_zastąpić, const string & co_wstawić, size_type odkąd_brać, size_type ile_znaków);`
- `string& replace(size_type odkąd_zastępować, size_type ile_zastąpić, const char * co_wstawić, size_type ile_znaków);`
- `string& replace(size_type odkąd_zastępować, size_type ile_zastąpić, const char * co_wstawić);`
- `string& replace(size_type odkąd_zastępować, size_type ile_zastąpić, size_type ile_znaków, char znak);`

string - metody - zamiana

- `string& replace(size_type odkąd_zastępować, size_type ile_zastąpić, const string & co_wstawić);`
- `string& replace(size_type odkąd_zastępować, size_type ile_zastąpić, const string & co_wstawić, size_type odkąd_brać, size_type ile_znaków);`
- `string& replace(size_type odkąd_zastępować, size_type ile_zastąpić, const char * co_wstawić, size_type ile_znaków);`
- `string& replace(size_type odkąd_zastępować, size_type ile_zastąpić, const char * co_wstawić);`
- `string& replace(size_type odkąd_zastępować, size_type ile_zastąpić, size_type ile_znaków, char znak);`

string - metody - zamiana

- `string& replace(size_type odkąd_zastępować, size_type ile_zastąpić, const string & co_wstawić);`
- `string& replace(size_type odkąd_zastępować, size_type ile_zastąpić, const string & co_wstawić, size_type odkąd_brać, size_type ile_znaków);`
- `string& replace(size_type odkąd_zastępować, size_type ile_zastąpić, const char * co_wstawić, size_type ile_znaków);`
- `string& replace(size_type odkąd_zastępować, size_type ile_zastąpić, const char * co_wstawić);`
- `string& replace(size_type odkąd_zastępować, size_type ile_zastąpić, size_type ile_znaków, char znak);`

string - metody - zamiana

- `string& replace(size_type odkąd_zastępować, size_type ile_zastąpić, const string & co_wstawić);`
- `string& replace(size_type odkąd_zastępować, size_type ile_zastąpić, const string & co_wstawić, size_type odkąd_brać, size_type ile_znaków);`
- `string& replace(size_type odkąd_zastępować, size_type ile_zastąpić, const char * co_wstawić, size_type ile_znaków);`
- `string& replace(size_type odkąd_zastępować, size_type ile_zastąpić, const char * co_wstawić);`
- `string& replace(size_type odkąd_zastępować, size_type ile_zastąpić, size_type ile_znaków, char znak);`

string - metody - porównywanie

- `int compare(const string& inny);`
- `int compare(size_type od_pozycji,
size_type ile_znaków, const string& inny);`
- `int compare(size_type od_pozycji,
size_type ile_znaków, const string& inny,
size_type poz_por, size_type ile_zn_por);`
- `int compare(const char* c_string);`
- `int compare(size_type od_pozycji,
size_type ile_znaków, const char* c_string,
size_type ile_zn_por = npos);`
- `== != < > <= >=`

string - metody - porównywanie

- `int compare(const string& inny);`
- `int compare(size_type od_pozycji, size_type ile_znaków, const string& inny);`
- `int compare(size_type od_pozycji, size_type ile_znaków, const string& inny, size_type poz_por, size_type ile_zn_por);`
- `int compare(const char* c_string);`
- `int compare(size_type od_pozycji, size_type ile_znaków, const char* c_string, size_type ile_zn_por = npos);`
- `== != < > <= >=`

string - metody - porównywanie

- `int compare(const string& inny);`
- `int compare(size_type od_pozycji, size_type ile_znaków, const string& inny);`
- `int compare(size_type od_pozycji, size_type ile_znaków, const string& inny, size_type poz_por, size_type ile_zn_por);`
- `int compare(const char* c_string);`
- `int compare(size_type od_pozycji, size_type ile_znaków, const char* c_string, size_type ile_zn_por = npos);`
- `== != < > <= >=`

string - metody - porównywanie

- `int compare(const string& inny);`
- `int compare(size_type od_pozycji, size_type ile_znaków, const string& inny);`
- `int compare(size_type od_pozycji, size_type ile_znaków, const string& inny, size_type poz_por, size_type ile_zn_por);`
- `int compare(const char* c_string);`
- `int compare(size_type od_pozycji, size_type ile_znaków, const char* c_string, size_type ile_zn_por = npos);`
- `== != < > <= >=`

string - metody - porównywanie

- `int compare(const string& inny);`
- `int compare(size_type od_pozycji, size_type ile_znaków, const string& inny);`
- `int compare(size_type od_pozycji, size_type ile_znaków, const string& inny, size_type poz_por, size_type ile_zn_por);`
- `int compare(const char* c_string);`
- `int compare(size_type od_pozycji, size_type ile_znaków, const char* c_string, size_type ile_zn_por = npos);`
- `== != < > <= >=`

string - metody - porównywanie

- `int compare(const string& inny);`
- `int compare(size_type od_pozycji, size_type ile_znaków, const string& inny);`
- `int compare(size_type od_pozycji, size_type ile_znaków, const string& inny, size_type poz_por, size_type ile_zn_por);`
- `int compare(const char* c_string);`
- `int compare(size_type od_pozycji, size_type ile_znaków, const char* c_string, size_type ile_zn_por = npos);`
- `== != < > <= >=`

string - metody - przypisanie

- `string & assign(string& inny_string);`
- `string & assign(string& inny_string, size_type odkąd, size_type ile_znaków);`
- `string & assign(char * c_string, size_type ile_znaków);`
- `string & assign(char * c_string);`
- `string & assign(size_type ile_znaków, char znak);`

string - metody - przypisanie

- `string & assign(string& inny_string);`
- `string & assign(string& inny_string, size_type odkąd, size_type ile_znaków);`
- `string & assign(char * c_string, size_type ile_znaków);`
- `string & assign(char * c_string);`
- `string & assign(size_type ile_znaków, char znak);`

string - metody - przypisanie

- `string & assign(string& inny_string);`
- `string & assign(string& inny_string, size_type odkąd, size_type ile_znaków);`
- `string & assign(char * c_string, size_type ile_znaków);`
- `string & assign(char * c_string);`
- `string & assign(size_type ile_znaków, char znak);`

string - metody - przypisanie

- `string & assign(string& inny_string);`
- `string & assign(string& inny_string, size_type odkąd, size_type ile_znaków);`
- `string & assign(char * c_string, size_type ile_znaków);`
- `string & assign(char * c_string);`
- `string & assign(size_type ile_znaków, char znak);`

string - metody - przypisanie

- `string & assign(string& inny_string);`
- `string & assign(string& inny_string, size_type odkąd, size_type ile_znaków);`
- `string & assign(char * c_string, size_type ile_znaków);`
- `string & assign(char * c_string);`
- `string & assign(size_type ile_znaków, char znak);`

string - metody - dopisywanie

- `string & append(string& inny_string);`
- `string & append(string& inny_string, size_type odkąd, size_type ile_znaków);`
- `string & append(char * c_string, size_type ile_znaków);`
- `string & append(char * c_string);`
- `string & append(size_type ile_znaków, char znak);`

string - metody - dopisywanie

- `string & append(string& inny_string);`
- `string & append(string& inny_string, size_type odkąd, size_type ile_znaków);`
- `string & append(char * c_string, size_type ile_znaków);`
- `string & append(char * c_string);`
- `string & append(size_type ile_znaków, char znak);`

string - metody - dopisywanie

- `string & append(string& inny_string);`
- `string & append(string& inny_string, size_type odkąd, size_type ile_znaków);`
- `string & append(char * c_string, size_type ile_znaków);`
- `string & append(char * c_string);`
- `string & append(size_type ile_znaków, char znak);`

string - metody - dopisywanie

- `string & append(string& inny_string);`
- `string & append(string& inny_string, size_type odkąd, size_type ile_znaków);`
- `string & append(char * c_string, size_type ile_znaków);`
- `string & append(char * c_string);`
- `string & append(size_type ile_znaków, char znak);`

string - metody - dopisywanie

- `string & append(string& inny_string);`
- `string & append(string& inny_string, size_type odkąd, size_type ile_znaków);`
- `string & append(char * c_string, size_type ile_znaków);`
- `string & append(char * c_string);`
- `string & append(size_type ile_znaków, char znak);`

string - metody

- `const char* c_str();`
- `const char* data();`
- `size_type copy(char * wsk_tablicy_znakow, size_type ile, size_type od_pozycji = 0);`
- `void swap(string & inny_string);`
- `istream & getline(istream wej, string schowek, char ogranicznik = '\n');`

string - metody

- `const char* c_str();`
- `const char* data();`
- `size_type copy(char * wsk_tablicy_znakow, size_type ile, size_type od_pozycji = 0);`
- `void swap(string & inny_string);`
- `istream & getline(istream wej, string schowek, char ogranicznik = '\n');`

string - metody

- `const char* c_str();`
- `const char* data();`
- `size_type copy(char * wsk_tablicy_znakow, size_type ile, size_type od_pozycji = 0);`
- `void swap(string & inny_string);`
- `istream & getline(istream wej, string schowek, char ogranicznik = '\n');`

string - metody

- `const char* c_str();`
- `const char* data();`
- `size_type copy(char * wsk_tablicy_znakow, size_type ile, size_type od_pozycji = 0);`
- `void swap(string & inny_string);`
- `istream & getline(istream wej, string schowek, char ogranicznik = '\\n');`

string - metody

- `const char* c_str();`
- `const char* data();`
- `size_type copy(char * wsk_tablicy_znakow, size_type ile, size_type od_pozycji = 0);`
- `void swap(string & inny_string);`
- `istream & getline(istream wej, string schowek, char ogranicznik = '\n');`

Zadania

Napisz algorytm który:

- 1 Napisz funkcję, która otrzymawszy jako argument obiekt klasy `string` pokaże nam kolejno wszystkie znaki. Każdy znak powinien być ujęty w nawias kwadratowy. Dla wejścia "Test" powinien się pojawić napis [T] [e] [s] [t]
- 2 Napisz funkcję, która otrzymawszy jako argument obiekt klasy `string` będący nazwą pliku, obetnie jego rozszerzenie. Dla wejścia "plik.html" powinna zwrócić napis "plik".
- 3 Napisz funkcję, która poleci użytkownikowi wprowadzić nazwę jakiegoś pliku tekstowego. Funkcja ta sprawdzi, czy w nazwie pliku użytkownik podał rozszerzenie ".txt". Jeśli nie ma rozszerzenia, to je doda, a jeśli jest inne, to je zmieni na "txt".
- 4 Napisz funkcję, która w stringu przesłanym jako argument zamieni znaki w taki sposób, że znaki na pozycjach parzystych będą napisane wielkimi literami, a nieparzystych małymi.