Kolokwium/

Generated by Doxygen 1.9.1

Chapter 1

Laboratoria 11: Kolokwium z roku 2021: PtrCStringVector

Materiał w tym roku jest podobny co w roku 2021, więc samodzielne zrobienie paczki daje duże szanse na zdanie tegorocznego kolokwium. Proszę spróbować zaimplementować to w oparciu o treść z pliku `PtrCStringVector.h` (alternatywnie w pliku `Documentation.pdf`).

Dobrze jakby Państwo spróbowali to kolokwium zrobić zamodzielnie, nie pytali się zbyt pochopnie innych. Jeśli Państwo sobie poradzą będzie to znak, że materiał na kolokwium w aktualnym roku Państwo w miarę ogarniają.

Sugeruję zrobić na jedno posiedzenie całą paczkę, licząc czas. Jeśli się komuś uda w dwie godziny wszystko, tzn. że jest na poziomie najlepszych osób sprzed dwóch lat.

1.1 Omówienie paczki:

O co chodzi w tym zadaniu znajduje się w krótkim `nagraniu`.

1.2 Dodatkowe wymagania:

Dla osób, którym się uda doprowadzić do sytuacji, że wszystkie testy przejdą proponuję dorobić następujące rzeczy:

1. Iterator (1 punkt z aktywności za zaimplementowanie + 1 punkt z aktywności za testy do tego)

Chapter 2

Todo List

Member FIRSTNAME
Uzupelnij swoje dane:

Member PtrCStringVector::free ()
sugeruje zaimplementowac, szczegoly w pliku naglowkowym

Member PtrCStringVector::operator& (const PtrCStringVector &anotherVector) const
zaimplementuj, szczegoly w pliku naglowkowym

Member PtrCStringVector::operator+ (const PtrCStringVector &anotherVector) const
zaimplementuj, szczegoly w pliku naglowkowym

Member PtrCStringVector::operator= (const PtrCStringVector &source)
zaimplementuj, szczegoly w pliku naglowkowym

Member PtrCStringVector::operator= (PtrCStringVector &&source)
zaimplementuj, szczegoly w pliku naglowkowym

Member PtrCStringVector::PtrCStringVector ()
zaimplementuj, szczegoly w pliku naglowkowym

Member PtrCStringVector::PtrCStringVector (const PtrCStringVector &source)
zaimplementuj, szczegoly w pliku naglowkowym

Member PtrCStringVector::push_back (const char *text2Add)
zaimplementuj, szczegoly w pliku naglowkowym

Member PtrCStringVector::reserve (std::size_t new_size)
sugeruje zaimplementowac, szczegoly w pliku naglowkowym

Member PtrCStringVector::~PtrCStringVector ()
zaimplementuj, szczegoly w pliku naglowkowym

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

Chapter 5

Class Documentation

5.1 PtrCStringVector Class Reference

Klasa PtrCStringVector, stanowiaca wektor wskaźników do niemodyfikowalnych tekstów. Wektor ten moze sie powiekszac o nowe elementy dokonujac kopiowania dotychczasowej zawartosci. Schemat znajduje się tutaj:

`#include <PtrCStringVector.h>`

Public Member Functions

- PtrCStringVector ()

  konstruktor domyslny, jego zadaniem jest ustawienie size_, capacity_ i data_ na brak elementow
- PtrCStringVector (const PtrCStringVector &source)

  *konstruktor kopiujacy, dokonujacy **gleboka kopie**, czyli nie tylko tablica wskaznikow na tekst musi zostac skopiowana ale rowniez wszystkie wskazywane teksty*
- ~PtrCStringVector ()

  destruktor, ktory musi koniecznie zwolnic pamiec i inne zasoby
- PtrCStringVector & operator= (PtrCStringVector &&source)

  operator przypisania, ktory ma za zadanie przeniesc zawartosc z obiektu zrodlowego
- PtrCStringVector & operator= (const PtrCStringVector &source)

  operator przypisania, ktory ma za zadanie skopiowac doglebnie tresc, analogicznie jak konstruktor kopiujacy PtrCStringVector(const PtrCStringVector&)
- void push_back (const char *text2Add)

  metoda, ktora skopiuje podany tekst i umiesci na koncu w Vectorze. W razie braku miejsca powinna dokonac powiekszenia kontenera.
- auto size () const

  Metoda zwracajaca aktualnie posiadana ilosc elementow w kontenerze.
- auto capacity () const

  Metoda zwracajaca informacje ile elementow zmiesci sie w zaalokowanej tablicy.
- char * operator[ ] (std::size_t index)

  operator indeksowania, ktory otrzymawszy indeks, zwroci wskaznik do tekstu znajdujacego sie na danej pozycji w kontenerze
- const char * operator[ ] (std::size_t index) const

  operator indeksowania, podobny do powyzszego operator[], ale zwraca `const char*` i jest metoda stala
- PtrCStringVector operator+ (const PtrCStringVector &anotherVector) const

  operator, ktory tworzy kontener zawierajacy wszystkie elementy z dowoch kontenerow (czyli dodaje kontenery)
- PtrCStringVector operator& (const PtrCStringVector &anotherVector) const

  operator& - ma za zadanie zwrocic nowo-utworzony kontener, ktory bedzie zawieral zawartosc obydwu kontenerow poprzez sklejenie tekstow na odpowiadajacych sobie pozycjach

Protected Member Functions

- void free ()

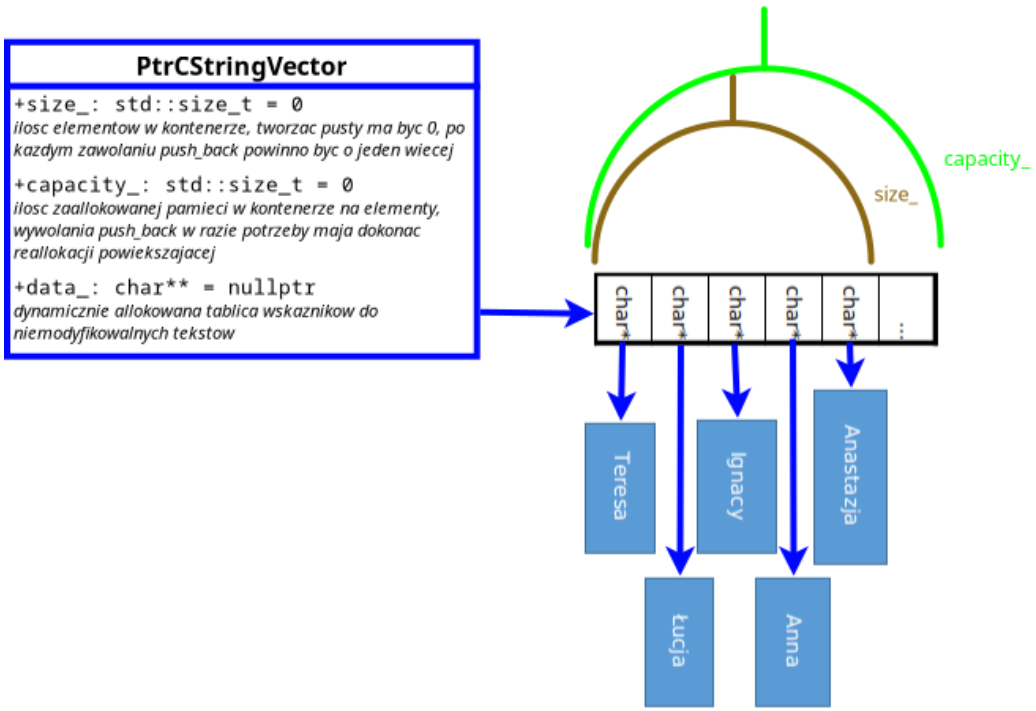
  metoda pomocnicza zwalniajaca wszystkie zasoby i zerujaca skladowe klasy
- void reserve (std::size_t new_size)

  metoda pomocnicza dokonujaca allokacji dynamicznej tablicy o okreslonym rozmiarze, nastepnie kopiujaca wszystkie elementy z dotychczasowej tablicy.

5.1.1 Detailed Description

Klasa PtrCStringVector, stanowiaca wektor wskaźników do niemodyfikowalnych tekstów. Wektor ten moze sie powiekszac o nowe elementy dokonujac kopiowania dotychczasowej zawartosci. Schemat znajduje się tutaj:

class PtrCStringVector



Note

Nie wolno uzywac typu `std::string`, nalezy uzyc tablicy char

Nie wolno uzywac `std::vector`, ma byc reczne zarzadzanie pamiecia

Można utworzyć dowolną ilość metod pomocniczych lub funkcji w pliku źródłowym. Niektóre metody pomocnicze są zadeklarowane, ale nie trzeba ich implementować.

Parameters

*size_*	ilosc elementow w kontenerze, tworzac pusty ma byc 0, po kazdym zawolaniu `push_back` powinno byc o jeden wiecej
*capacity_*	ilosc zaallokowanej pamieci w kontenerze na elementy, wywolania `push_back` w razie potrzeby maja dokonac reallokacji powiekszajacej
*data_*	dynamicznie allokowana tablica wskaznikow do niemodyfikowalnych tekstow

Definition at line 28 of file PtrCStringVector.h.

5.1.2 Constructor & Destructor Documentation

5.1.2.1 PtrCStringVector() [1/2]

```
PtrCStringVector::PtrCStringVector ( )
```

konstruktor domyslny, jego zadaniem jest ustawienie size_, capacity_ i data_ na brak elementow

Todo zaimplementuj, szczegoly w pliku naglowkowym

Definition at line 11 of file PtrCStringVector.cpp.

5.1.2.2 PtrCStringVector() [2/2]

```
PtrCStringVector::PtrCStringVector (
            const PtrCStringVector & source )
```

konstruktor kopiujacy, dokonujacy **gleboka kopie**, czyli nie tylko tablica wskaznikow na tekst musi zostac skopiowana ale rowniez wszystkie wskazywane teksty

Parameters

source	- kontener zrodkowy, z ktorego musza byc skopiowane wszystkie dane

Todo zaimplementuj, szczegoly w pliku naglowkowym

Definition at line 17 of file PtrCStringVector.cpp.

5.1.2.3 ~PtrCStringVector()

```
PtrCStringVector::~PtrCStringVector ( )
```

destruktor, ktory musi koniecznie zwolnic pamiec i inne zasoby

Todo zaimplementuj, szczegoly w pliku naglowkowym

Definition at line 23 of file PtrCStringVector.cpp.

5.1.3 Member Function Documentation

5.1.3.1 capacity()

```
auto PtrCStringVector::capacity ( ) const  [inline]
```

Metoda zwracajaca informacje ile elementow zmiesci sie w zaalokowanej tablicy.

Returns

wartosc `capacity_`

Definition at line 80 of file PtrCStringVector.h.

5.1.3.2 free()

```
void PtrCStringVector::free ( )  [protected]
```

metoda pomocnicza zwalniajaca wszystkie zasoby i zerujaca skladowe klasy

Todo sugeruje zaimplementowac, szczegoly w pliku naglowkowym

Definition at line 72 of file PtrCStringVector.cpp.

5.1.3.3 operator&()

```
PtrCStringVector PtrCStringVector::operator& (
            const PtrCStringVector & anotherVector ) const
```

operator& - ma za zadanie zwrocic nowo-utworzony kontener, ktory bedzie zawieral zawartosc obydwu kontenerow poprzez sklejenie tekstow na odpowiadajacych sobie pozycjach

Parameters

anotherVector	kontener z ktorego maja byc doklejone elementy do tych z `*this`

Returns

Nowo-utworzony kontener zawierajacy wszystkie elementy z `*this` i `anotherVector` posklejane na odpowiadajacych sobie pozycjach

W razie braku odpowiadajacych sobie elementow (gdy rozmiary kontenerow sa inne) nalezy od miejsca gdzie elementy ma tylko jeden wziac element tylko z kontenera, ktory go posiada. Nie nalezy doklejac czegokolwiek laczac napisy (zadnej spacji, srednika itp). Przykladowo mamy dwa kontenery (zbieznosc imion przypadkowa):

1. `PtrCStringVector kobietyWGrupie;`, ktory zawiera teksty:
   (a) "Anastazja"
   (b) "Bonifia"
   (c) "Cecylia"
2. `PtrCStringVector mezczyzniWGrupie;`, ktory zawiera teksty:
   (a) "Ambrozy"
   (b) "Bazyli"
   (c) "Cezary"
   (d) "Dionizy"
   (e) "Elohim"
3. Wynikowy `kobietyWGrupie & mezczyzniWGrupie` powinien zawierac:
   (a) "AnastazjaAmbrozy"
   (b) "BonifiaBazyli"
   (c) "CecyliaCezary"
   (d) "Dionizy"
   (e) "Elohim"

Todo zaimplementuj, szczegoly w pliku naglowkowym

Definition at line 66 of file PtrCStringVector.cpp.

5.1.3.4 operator+()

```
PtrCStringVector PtrCStringVector::operator+ (
            const PtrCStringVector & anotherVector ) const
```

operator, ktory tworzy kontener zawierajacy wszystkie elementy z dowoch kontenerow (czyli dodaje kontenery)

Parameters

anotherVector	kontener z ktorego maja byc dorzucone elementy

Returns

nowo-utworzony kontener zawierajacy wszystkie elementy z `*this` i `anotherVector`

wpierw beda skopiowane elementy z `*this`, nastepnie wszystkie z `anotherVector`

Todo zaimplementuj, szczegoly w pliku naglowkowym

Definition at line 49 of file PtrCStringVector.cpp.

5.1.3.5 operator=() [1/2]

```
PtrCStringVector & PtrCStringVector::operator= (
            const PtrCStringVector & source )
```

operator przypisania, ktory ma za zadanie skopiowac doglebnie tresc, analogicznie jak konstruktor kopiujacy PtrCStringVector(const PtrCStringVector&)

Note

prosze sie upewnic, ze zadziala przypisanie na samego siebie:

```
PtrCStringVector a;
PtrCStringVector& b = a;
a = b;
```

prosze sie upewnic, ze zadziala przypisanie kaskadowe:

```
PtrCStringVector a, b, c;
a = b = c;
```

Operator przypisania powinien **zwolnic pamiec** w razie potrzeby, aby nie dopuscic do wyciekow pamieci.

Todo zaimplementuj, szczegoly w pliku naglowkowym

Definition at line 29 of file PtrCStringVector.cpp.

5.1.3.6 operator=() [2/2]

```
PtrCStringVector & PtrCStringVector::operator= (
            PtrCStringVector && source )
```

operator przypisania, ktory ma za zadanie przeniesc zawartosc z obiektu zrodlowego

Note

prosze sie upewnic, ze zadziala przypisanie na samego siebie:

```
PtrCStringVector a;
PtrCStringVector& b = a;
a = std::move(b);
```

Operator przypisania przenoszacy powinien **zwolnic dotychczasowa pamiec**, aby nie dopuscic do wyciekow pamieci. Powinien tez zostawic obiekt zrodlowy w stanie jak po zawolaniu konstruktora domyslnego.

Todo zaimplementuj, szczegoly w pliku naglowkowym

Definition at line 36 of file PtrCStringVector.cpp.

5.1.3.7 operator[]() [1/2]

```
char* PtrCStringVector::operator[] (
            std::size_t index )
```

operator indeksowania, ktory otrzymawszy indeks, zwroci wskaznik do tekstu znajdujacego sie na danej pozycji w kontenerze

Parameters

index	elementu tekstowego w kontenerze

Exceptions

std::out_of_range	w razie, gdy `index >= size_`

5.1.3.8 operator[]() [2/2]

```
const char* PtrCStringVector::operator[] (
            std::size_t index ) const
```

operator indeksowania, podobny do powyzszego operator[], ale zwraca `const char*` i jest metoda stala

Exceptions

std::out_of_range	w razie, gdy `index >= size_`

5.1.3.9 push_back()

```
void PtrCStringVector::push_back (
            const char * text2Add )
```

metoda, ktora skopiuje podany tekst i umiesci na koncu w Vectorze. W razie braku miejsca powinna dokonac powiekszenia kontenera.

Parameters

text2Add	- tekst do skopiowania doglebnie (na nowa dynamiczna pamiec)

Postcondition

Dodany tekst zostanie skopiowany i umieszczony na koncu kontenera. W razie potrzeby tablica wskaznikow powinna byc powiekszona.

Todo zaimplementuj, szczegoly w pliku naglowkowym

Definition at line 43 of file PtrCStringVector.cpp.

5.1.3.10 reserve()

```
void PtrCStringVector::reserve (
            std::size_t new_size )  [protected]
```

metoda pomocnicza dokonujaca allokacji dynamicznej tablicy o okreslonym rozmiarze, nastepnie kopiujaca wszystkie elementy z dotychczasowej tablicy.

Note

nalezy pamietac o zwalnianiu zasobow

dla uproszczenia zakladamy, ze metoda ta jedynie zwieksza zaalokowana pamiec, nie zmniejsza

Todo sugeruje zaimplementowac, szczegoly w pliku naglowkowym

Definition at line 77 of file PtrCStringVector.cpp.

5.1.3.11 size()

```
auto PtrCStringVector::size ( ) const  [inline]
```

Metoda zwracajaca aktualnie posiadana ilosc elementow w kontenerze.

Returns

wartosc `size_`

Definition at line 73 of file PtrCStringVector.h.

The documentation for this class was generated from the following files:

- PtrCStringVector.h
- PtrCStringVector.cpp

Chapter 6

File Documentation

6.1 CMakeLists.txt File Reference

6.2 main.cpp File Reference

```
#include <iostream>
#include "PtrCStringVector.h"
```

Include dependency graph for main.cpp:

Functions

- void validateStudentsInfo ()
- int main ()
- constexpr size_t compileTimeStrlen (const char *text) noexcept
- constexpr size_t compileTimeCountFirstDigits (const char *text) noexcept
- constexpr bool compileTimeIsDigit (const char *text) noexcept
- constexpr bool compileTimeContains (const char *text, char letter) noexcept

Variables

- constexpr const char *const FIRSTNAME = ""
- constexpr const char *const SURNAME = ""
- constexpr const char *const MAIL = ""
- constexpr const char *const BOOK_ID = ""
- constexpr const char *const TEACHER_MAIL = "bazior[at]agh.edu.pl"

6.2.1 Function Documentation

6.2.1.1 compileTimeContains()

```
constexpr bool compileTimeContains (
            const char * text,
            char letter )  [inline], [constexpr], [noexcept]
```

Definition at line 39 of file main.cpp.

Here is the caller graph for this function:

6.2.1.2 compileTimeCountFirstDigits()

```
constexpr size_t compileTimeCountFirstDigits (
            const char * text )  [inline], [constexpr], [noexcept]
```

Definition at line 29 of file main.cpp.

Here is the caller graph for this function:

6.2.1.3 compileTimeIsDigit()

```
constexpr bool compileTimeIsDigit (
            const char * text )  [inline], [constexpr], [noexcept]
```

Definition at line 34 of file main.cpp.

Here is the call graph for this function: Here is the caller graph for this function:

6.2.1.4 compileTimeStrlen()

```
constexpr size_t compileTimeStrlen (
            const char * text )  [inline], [constexpr], [noexcept]
```

Definition at line 24 of file main.cpp.

Here is the caller graph for this function:

6.2.1.5 main()

```
int main ( )
```

Definition at line 15 of file main.cpp.

Here is the call graph for this function:

6.2.1.6 validateStudentsInfo()

```
void validateStudentsInfo ( )
```

Definition at line 47 of file main.cpp.

Here is the call graph for this function: Here is the caller graph for this function:

6.2.2 Variable Documentation

6.2.2.1 BOOK_ID

```
constexpr const char* const BOOK_ID = ""  [constexpr]
```

Definition at line 9 of file main.cpp.

6.2.2.2 FIRSTNAME

```
constexpr const char* const FIRSTNAME = ""  [constexpr]
```

Todo Uzupelnij swoje dane:

Definition at line 6 of file main.cpp.

6.2.2.3 MAIL

```
constexpr const char* const MAIL = ""  [constexpr]
```

Definition at line 8 of file main.cpp.

6.2.2.4 SURNAME

```
constexpr const char* const SURNAME = ""  [constexpr]
```

Definition at line 7 of file main.cpp.

6.2.2.5 TEACHER_MAIL

```
constexpr const char* const TEACHER_MAIL = "bazior[at]agh.edu.pl"  [constexpr]
```

Definition at line 10 of file main.cpp.

6.3 PtrCStringVector.cpp File Reference

```
#include <functional>
#include <algorithm>
#include <string>
#include <cstring>
#include <stdexcept>
#include <utility>
#include "PtrCStringVector.h"
```

Include dependency graph for PtrCStringVector.cpp:

6.4 PtrCStringVector.h File Reference

W ramach kolokwium trzeba zaimplementowac wszystkie opisane metody klasy PtrCStringVector. Do ponizszych metod **sa testy** w pliku PtrCStringVectorTests.cpp.

```
#include <cstddef>
```

Include dependency graph for PtrCStringVector.h: This graph shows which files directly or indirectly include this file:

Classes

- class PtrCStringVector

  Klasa PtrCStringVector, stanowiaca wektor wskaźników do niemodyfikowalnych tekstów. Wektor ten moze sie powiekszac o nowe elementy dokonujac kopiowania dotychczasowej zawartosci. Schemat znajduje się tutaj:

6.4.1 Detailed Description

W ramach kolokwium trzeba zaimplementowac wszystkie opisane metody klasy PtrCStringVector. Do ponizszych metod **sa testy** w pliku PtrCStringVectorTests.cpp.

Date

9 czerwca 2021

6.5 README.md File Reference

Index