

Kolokwium/

Generated by Doxygen 1.9.1



<b>1 Kolokwium z Podstaw Programowania 2 dnia 7 VI 2021</b>	<b>1</b>
1.0.1 Zasady ogólne: . . . . .	1
<b>2 Todo List</b>	<b>3</b>
<b>3 Class Index</b>	<b>5</b>
3.1 Class List . . . . .	5
<b>4 File Index</b>	<b>7</b>
4.1 File List . . . . .	7
<b>5 Class Documentation</b>	<b>9</b>
5.1 PtrCStringVector Class Reference . . . . .	9
5.1.1 Detailed Description . . . . .	10
5.1.2 Constructor & Destructor Documentation . . . . .	10
5.1.2.1 PtrCStringVector() [1/2] . . . . .	10
5.1.2.2 PtrCStringVector() [2/2] . . . . .	11
5.1.2.3 ~PtrCStringVector() . . . . .	11
5.1.3 Member Function Documentation . . . . .	11
5.1.3.1 capacity() . . . . .	11
5.1.3.2 copy() . . . . .	12
5.1.3.3 free() . . . . .	12
5.1.3.4 move() . . . . .	12
5.1.3.5 operator&() . . . . .	12
5.1.3.6 operator+() . . . . .	14
5.1.3.7 operator=() [1/2] . . . . .	15
5.1.3.8 operator=() [2/2] . . . . .	16
5.1.3.9 operator[]() [1/2] . . . . .	16
5.1.3.10 operator[]() [2/2] . . . . .	17
5.1.3.11 push_back() . . . . .	18
5.1.3.12 reserve() . . . . .	19
5.1.3.13 size() . . . . .	19
<b>6 File Documentation</b>	<b>21</b>
6.1 CMakeLists.txt File Reference . . . . .	21
6.1.1 Function Documentation . . . . .	21
6.1.1.1 add_custom_target() . . . . .	21
6.1.1.2 cmake_minimum_required() . . . . .	21
6.2 main.cpp File Reference . . . . .	22
6.2.1 Function Documentation . . . . .	22
6.2.1.1 compileTimeCountFirstDigits() . . . . .	23
6.2.1.2 compileTimeIsDigit() . . . . .	23
6.2.1.3 compileTimeStrlen() . . . . .	24
6.2.1.4 main() . . . . .	24

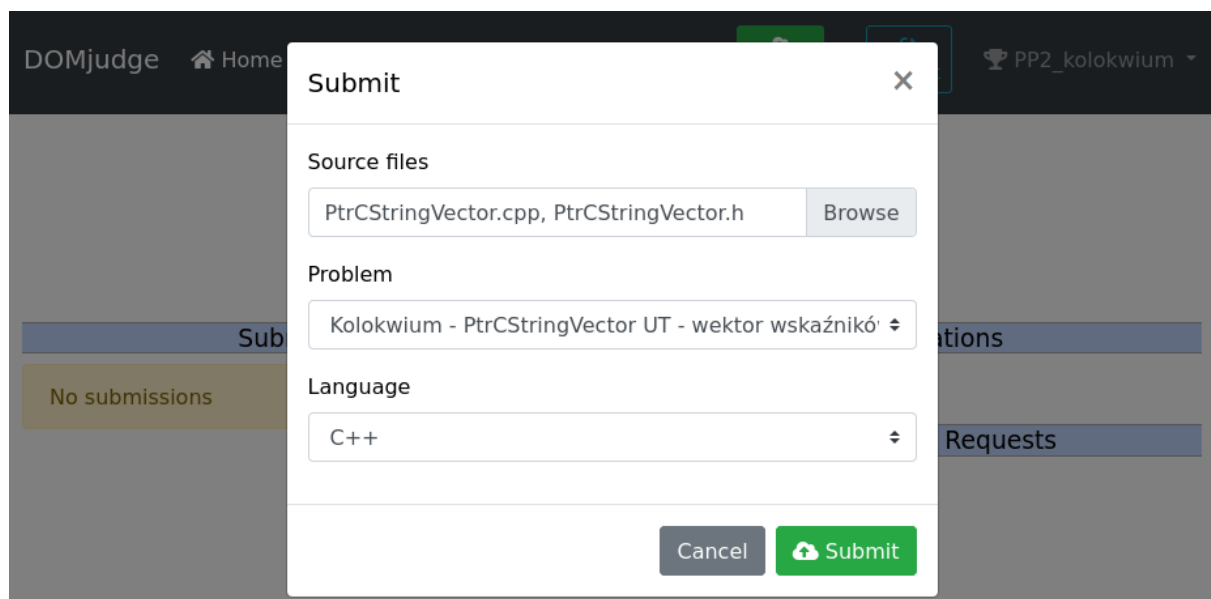
6.2.1.5 validateStudentsInfo() . . . . .	24
6.2.2 Variable Documentation . . . . .	24
6.2.2.1 BOOK_ID . . . . .	25
6.2.2.2 FIRSTNAME . . . . .	25
6.2.2.3 MAIL . . . . .	25
6.2.2.4 SURNAME . . . . .	25
6.2.2.5 teacherMail . . . . .	25
6.3 PtrCStringVector.cpp File Reference . . . . .	26
6.4 PtrCStringVector.h File Reference . . . . .	26
6.4.1 Detailed Description . . . . .	27
<b>Index</b>	<b>29</b>

## Chapter 1

# Kolokwium z Podstaw Programowania 2 dnia 7 VI 2021

### 1.0.1 Zasady ogólne:

1. Zanim cokolwiek Państwo zrobia prosze o uzupełnienie swoich danych w pliku: programmerDetails.h
2. Kolokwium nie da się nie zdać - jest traktowane jako punkty, które się sumują do reszty, nie będzie poprawki grupowej!
3. W trakcie kolokwium należy mieć włączona kamerkę i nie mieć słuchawek, kto będzie bez kamery to o 1/3 punktów mniej.
4. Wyslana paczka ma sie bezwzględnie kompilowac na systemie Linux.
  - Jak ktos nie ma linuxa moze uzyc narzedzia: <http://administrare.kis.agh.edu.pl:12345> (konieczny VPN AGH) i na nim zarejestrowac sie, ale jako nick powinien byc **numer albumu**, aczkolwiek prosze tez uzupelnic pozostale dane (imie, nazwisko, mail).
  - Aby wyslac zadanie nalezy wybrac konkurs (*kolokwium pp2*), problem (*kolokwium*), oraz jezyk programowania (*c++*), prosze zalaczyc obydwa pliki [PtrCStringVector.h](#), [PtrCStringVector.cpp](#), jak na obrazku:



5. Kolokwium z założenia będzie sprawdzane automatycznie, przez testy jednostkowe dostępne w pliku: `Ptr↔CStringVectorTests.cpp`, dlatego poza kompilowaniem proszę aby nie crashowało na żadnym teście, jeśli tak się dzieje to brane pod uwagę jest tylko tyle testów ile ich przechodzi do momentu crasha.
  6. Mam program antyplagiatowy, dlatego proszę pracować samodzielnie! Osoby które udostępniają swoje rozwiązania również będą miały kary!
  7. \*Dobrze jakby nie było warningów kompilacji (flagi: `-Wall -Wextra -pedantic -Werror`, a dla hardcorów jeszcze: `-Weffc++`)
  8. Punkty będą odejmowane za wycieki pamięci.
  9. Zakres materiału: wykłady [1, 10]  
Szczegółowa treść kolokwium znajduje Państwo w opisie metod klasy: [PtrCStringVector](#)
-

## Chapter 2

### Todo List

**Member `FIRSTNAME`**

Uzupelnij swoje dane:

**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::operator[] (size_t index)`**

zaimplementuj, szczegoly w pliku naglowkowym

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

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::~~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:

#### [PtrCStringVector](#)

Klasa [PtrCStringVector](#), stanowiaca wektor wskaźników do niemodyfikowalnych tekstów. Wektor ten moze sie powiekszac o nowe elementy dokonujac kopiowania dotychczasowej zawartosci .

[9](#)



## Chapter 4

# File Index

### 4.1 File List

Here is a list of all files with brief descriptions:

<a href="#">main.cpp</a>	22
<a href="#">PtrCStringVector.cpp</a>	26
<a href="#">PtrCStringVector.h</a>	
W ramach kolokwium trzeba zaimplementowac wszystkie opisane metody klasy <a href="#">PtrCStringVector</a> .	
Do ponizszych metod <b>sa testy</b> w pliku PtrCStringVectorTests.cpp	26



## Chapter 5

# Class Documentation

### 5.1 PtrCStringVector Class Reference

Klasa [PtrCStringVector](#), stanowiaca wektor wskaźników do niemodyfikowalnych tekstów. Wektor ten może się powiększać o nowe elementy dokonując kopiowania dotychczasowej zawartości.

```
#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. Przed dodaniem rozmiar zaalokowanej tablicy powinien zostac zwiekszony.*
- 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\[\]](#) (size\_t index)  
*operator indeksowania, ktory otrzymawszy indeks zwroci wskaznik do tekstu znajdujacego sie na danej pozycji w kontenerze*
- const char \* [operator\[\]](#) (size\_t index) const  
*operator indeksowania, podobny do powyzzszego [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 zwalnijaca wszystkie zasoby i zerujaca skladowe klasy*
- void `copy` (const `PtrCStringVector` &other)  
*metoda pomocnicza kopiujaca z obiektu zrodlowego na \*this.*
- void `move` (`PtrCStringVector` &other)  
*metoda pomocnicza przenoszaca zawartosc z obiektu zrodlowego na \*this.*
- void `reserve` (size\_t new\_size)  
*metoda pomocnicza dokonujaca alokacji 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.

class `PtrCStringVector`

#### Note

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

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

Zmiany pliku naglowkowego nie powinny byc konieczne, prosze probowac zmienic jedynie w pliku zrodlowym `PtrCStringVector.cpp`, wyjatkowo zamiast metod prosze robic funkcje

#### Parameters

<code>size_</code>	ilosc elementow w kontenerze, tworzac pusty ma byc 0, po kazdym zawolaniu <code>push_back</code> powinno byc o jeden wiecej
<code>capacity_</code>	ilosc zaalokowanej pamieci w kontenerze na elementy, wywolania <code>push_back</code> w razie potrzeby maja dokonac realokacji powiekszajacej
<code>data_</code>	dynamicznie alokowana tablica wskaznikow do niemodyfikowalnych tekstow

Definition at line 25 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 12 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

<code>source</code>	- kontener zrodkowy, z ktorego musza byc skopiowane wszystkie dane
---------------------	--

**Todo** zaimplementuj, szczegoly w pliku naglowkowym

Definition at line 20 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 26 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 78 of file PtrCStringVector.h.

### 5.1.3.2 copy()

```
void PtrCStringVector::copy (
    const PtrCStringVector & other ) [protected]
```

metoda pomocnicza kopiujaca z obiektu zrodlowego na \*this.

#### Note

nalezy pamietac o zwalnianiu zasobow

Definition at line 125 of file PtrCStringVector.cpp.

### 5.1.3.3 free()

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

metoda pomocnicza zwalnijaca wszystkie zasoby i zerujaca skladowe klasy

Definition at line 118 of file PtrCStringVector.cpp.

### 5.1.3.4 move()

```
void PtrCStringVector::move (
    PtrCStringVector & other ) [protected]
```

metoda pomocnicza przenoszaca zawartosc z obiektu zrodlowego na \*this.

#### Note

nalezy pamietac o zwalnianiu zasobow

nalezy pamietac, aby zostawic obiekt zrodlowy w stanie jak po wywoalniu konstruktora bezargumentowego

Definition at line 130 of file PtrCStringVector.cpp.

### 5.1.3.5 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

<code>anotherVector</code>	kontener z ktorego maja byc doklejone elementy do tych z <code>*this</code>
----------------------------	---

## Returns

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

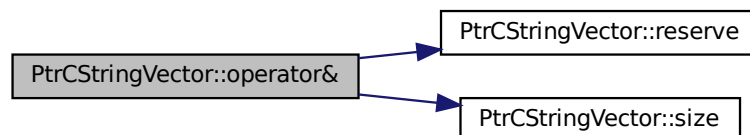
W razie braku odpowiadajacych sobie elementow (gdy rozmiary kontenerow sa inne) nalezy wziac element tylko z kontenera, ktory go posiada. Nie nalezy doklejac czegokolwiek laczac napisy (zadnej spacji, srednika itp). Przykladowo mamy dwa kontenery (zbiezność imion przypadkowa):

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 91 of file `PtrCStringVector.cpp`.

Here is the call graph for this function:



#### 5.1.3.6 operator+()

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

operator, który tworzy kontener zawierający wszystkie elementy z dwóch kontenerów (czyli dodaje kontenery)

## Parameters

<code>anotherVector</code>	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 z `anotherVector`

**Todo** zaimplementuj, szczegoly w pliku naglowkowym

Definition at line 64 of file `PtrCStringVector.cpp`.

Here is the call graph for this function:



## 5.1.3.7 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. Mozna uzyc **idiomu copy&swap**.

**Todo** zaimplementuj, szczegoly w pliku naglowkowym

Definition at line 32 of file `PtrCStringVector.cpp`.

### 5.1.3.8 operator=() [2/2]

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

operator przypisania, który ma za zadanie przenieść zawartość z obiektu źródłowego

#### Note

prosze sie upewnic, że zadziała przypisanie na samego siebie:

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

Operator przypisania przenoszący powinien **zwolnić dotychczasową pamięć**, aby nie dopuścić do wycieków pamięci. Powinien też zostawić obiekt źródłowy w stanie jak po zawołaniu konstruktora domyślnego.

**Todo** zaimplementuj, szczegóły w pliku nagłówkowym

Definition at line 42 of file PtrCStringVector.cpp.

### 5.1.3.9 operator[]() [1/2]

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

operator indeksowania, który otrzymawszy indeks zwróci wskaźnik do tekstu znajdującego się na danej pozycji w kontenerze

#### Parameters

<i>index</i>	elementu tekstowego w kontenerze
--------------	----------------------------------

#### Exceptions

w	razie podania
<i>std::out_of_range</i>	w razie, gdy <i>index</i> >= <i>size_</i>

**Todo** zaimplementuj, szczegóły w pliku nagłówkowym

Definition at line 73 of file PtrCStringVector.cpp.

#### 5.1.3.10 operator[]() [2/2]

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

operator indeksowania, podobny do powyższego [operator\[\]](#), ale zwraca `const char*` i jest metoda stała

### Exceptions

<i>jeżeli</i>	index jest poza zakresem zwraca pusty tekst ""
---------------	--

**Todo** zaimplementuj, szczegoly w pliku naglowkowym

Definition at line 79 of file PtrCStringVector.cpp.

#### 5.1.3.11 push\_back()

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

metoda, która skopiuje podany tekst i umieści na końcu w Vectorze. Przed dodaniem rozmiar zaalokowanej tablicy powinien zostać zwiększony.

### Parameters

<i>text2Add</i>	- tekst do skopiowania doglebnie (na nowa dynamiczna pamiec)
-----------------	--

### Postcondition

Dodany tekst zostanie skopiowany i umieszczony na końcu kontenera. W razie potrzeby tablica wskaźników powinna być powiększona.

**Todo** zaimplementuj, szczegoly w pliku naglowkowym

Definition at line 53 of file PtrCStringVector.cpp.

Here is the caller graph for this function:



### 5.1.3.12 reserve()

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

metoda pomocnicza dokonujaca alokacji 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 zwielsza zaalokowana pamiec, nie zmniejsza

Definition at line 139 of file PtrCStringVector.cpp.

Here is the caller graph for this function:



### 5.1.3.13 size()

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

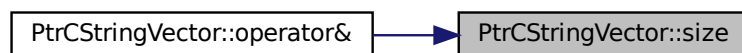
Metoda zwracajaca aktualnie posiadana ilosc elementow w kontenerze.

#### Returns

wartosc `size_`

Definition at line 71 of file PtrCStringVector.h.

Here is the caller graph for this function:



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

#### Functions

- [cmake\\_minimum\\_required](#) (VERSION 3.19) project(kolokwium) set(CMAKE\_CXX\_STANDARD 14) add\_subdirectory(google\_test) add\_executable(\$
- [main](#) cpp [PtrCStringVector](#) cpp [add\\_custom\\_target](#) (run COMMAND \${PROJECT\_NAME} DEPENDS \${PROJECT\_NAME} WORKING\_DIRECTORY \${CMAKE\_CURRENT\_BINARY\_DIR}) add\_custom\_target(valgrind\_\$

#### 6.1.1 Function Documentation

##### 6.1.1.1 add\_custom\_target()

```
main cpp PtrCStringVector cpp add_custom_target (
    run COMMAND ${PROJECT_NAME} DEPENDS ${PROJECT_NAME} WORKING_DIRECTORY ${CMAKE_
CURRENT_BINARY_DIR} )
```

Definition at line 8 of file CMakeLists.txt.

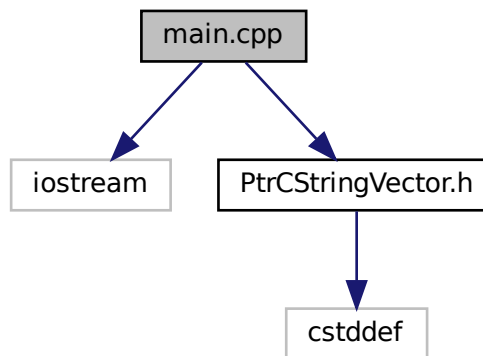
##### 6.1.1.2 cmake\_minimum\_required()

```
cmake_minimum_required (
    VERSION 3. 19 )
```

Definition at line 1 of file CMakeLists.txt.

## 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 `compileTimelsDigit` (const char \*text) 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 `teacherMail` = "bazior[at]agh.edu.pl"

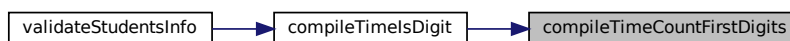
### 6.2.1 Function Documentation

### 6.2.1.1 compileTimeCountFirstDigits()

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

Definition at line 52 of file main.cpp.

Here is the caller graph for this function:

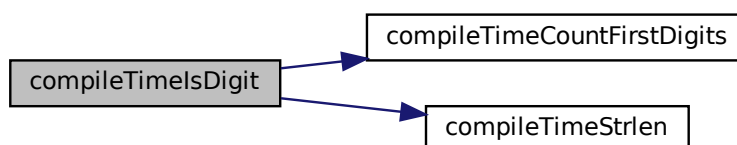


### 6.2.1.2 compileTimeIsDigit()

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

Definition at line 57 of file main.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

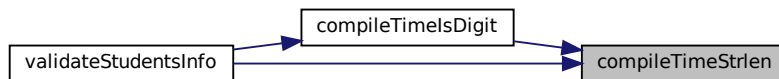


### 6.2.1.3 compileTimeStrlen()

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

Definition at line 47 of file main.cpp.

Here is the caller graph for this function:



### 6.2.1.4 main()

```
int main ( )
```

Definition at line 40 of file main.cpp.

### 6.2.1.5 validateStudentsInfo()

```
void validateStudentsInfo ( )
```

Definition at line 62 of file main.cpp.

Here is the call 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 34 of file main.cpp.

### 6.2.2.2 FIRSTNAME

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

**Todo** Uzupełnij swoje dane:

Definition at line 31 of file main.cpp.

### 6.2.2.3 MAIL

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

Definition at line 33 of file main.cpp.

### 6.2.2.4 SURNAME

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

Definition at line 32 of file main.cpp.

### 6.2.2.5 teacherMail

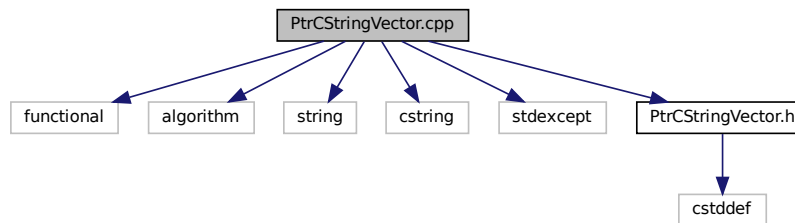
```
constexpr const char* const teacherMail = "bazior[at]agh.edu.pl" [constexpr]
```

Definition at line 35 of file main.cpp.

## 6.3 PtrCStringVector.cpp File Reference

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

Include dependency graph for PtrCStringVector.cpp:

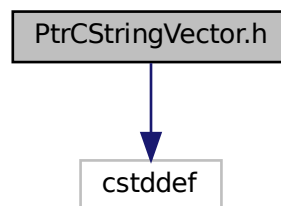


## 6.4 PtrCStringVector.h File Reference

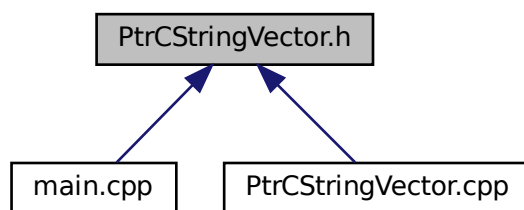
W ramach kolokwium trzeba zaimplementowac wszystkie opisane metody klasy `PtrCStringVector`. Do ponizszych metod **sa testy** w pliku `PtrCStringVectorTests.cpp`.

```
#include <cstddef>
```

Include dependency graph for PtrCStringVector.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [PtrCStringVector](#)

*Klasa [PtrCStringVector](#), stanowiąca wektor wskaźników do niemodyfikowalnych tekstów. Wektor ten może się powiększać o nowe elementy dokonując kopiowania dotychczasowej zawartości.*

### 6.4.1 Detailed Description

W ramach kolokwium trzeba zaimplementować wszystkie opisane metody klasy [PtrCStringVector](#). Do poniższych metod **sa testy** w pliku `PtrCStringVectorTests.cpp`.

#### Date

9 czerwca 2021





# Index

- [~PtrCStringVector](#)
    - [PtrCStringVector](#), [11](#)
- [add\\_custom\\_target](#)
  - [CMakeLists.txt](#), [21](#)
- [BOOK\\_ID](#)
  - [main.cpp](#), [24](#)
- [capacity](#)
  - [PtrCStringVector](#), [11](#)
- [cmake\\_minimum\\_required](#)
  - [CMakeLists.txt](#), [21](#)
- [CMakeLists.txt](#), [21](#)
  - [add\\_custom\\_target](#), [21](#)
  - [cmake\\_minimum\\_required](#), [21](#)
- [compileTimeCountFirstDigits](#)
  - [main.cpp](#), [22](#)
- [compileTimeIsDigit](#)
  - [main.cpp](#), [23](#)
- [compileTimeStrlen](#)
  - [main.cpp](#), [23](#)
- [copy](#)
  - [PtrCStringVector](#), [11](#)
- [FIRSTNAME](#)
  - [main.cpp](#), [25](#)
- [free](#)
  - [PtrCStringVector](#), [12](#)
- [MAIL](#)
  - [main.cpp](#), [25](#)
- [main](#)
  - [main.cpp](#), [24](#)
- [main.cpp](#), [22](#)
  - [BOOK\\_ID](#), [24](#)
  - [compileTimeCountFirstDigits](#), [22](#)
  - [compileTimeIsDigit](#), [23](#)
  - [compileTimeStrlen](#), [23](#)
  - [FIRSTNAME](#), [25](#)
  - [MAIL](#), [25](#)
  - [main](#), [24](#)
  - [SURNAME](#), [25](#)
  - [teacherMail](#), [25](#)
  - [validateStudentsInfo](#), [24](#)
- [move](#)
  - [PtrCStringVector](#), [12](#)
- [operator+](#)
  - [PtrCStringVector](#), [13](#)
- [operator=](#)
  - [PtrCStringVector](#), [15](#)
- [operator&](#)
  - [PtrCStringVector](#), [12](#)
- [operator\[\]](#)
  - [PtrCStringVector](#), [16](#)
- [PtrCStringVector](#), [9](#)
  - [~PtrCStringVector](#), [11](#)
  - [capacity](#), [11](#)
  - [copy](#), [11](#)
  - [free](#), [12](#)
  - [move](#), [12](#)
  - [operator+](#), [13](#)
  - [operator=](#), [15](#)
  - [operator&](#), [12](#)
  - [operator\[\]](#), [16](#)
  - [PtrCStringVector](#), [10](#)
  - [push\\_back](#), [18](#)
  - [reserve](#), [18](#)
  - [size](#), [19](#)
- [PtrCStringVector.cpp](#), [26](#)
- [PtrCStringVector.h](#), [26](#)
- [push\\_back](#)
  - [PtrCStringVector](#), [18](#)
- [reserve](#)
  - [PtrCStringVector](#), [18](#)
- [size](#)
  - [PtrCStringVector](#), [19](#)
- [SURNAME](#)
  - [main.cpp](#), [25](#)
- [teacherMail](#)
  - [main.cpp](#), [25](#)
- [validateStudentsInfo](#)
  - [main.cpp](#), [24](#)