

Politechnika Śląska  
Wydział Informatyki, Elektroniki i Informatyki

# Programowanie Komputerów 3

Rezerwacje w kinie

---

autor	Daniel Wiszowaty
prowadzący	mgr. Grzegorz Wojciech Kwiatkowski
rok akademicki	2020/2021
kierunek	informatyka
rodzaj studiów	SSI
semestr	3
sekcja	21
termin oddania sprawozdania	2020-11-10
git	polsl-aei-pk3 / aa44cccc-gr22-repo

---

# 1 Treść zadania

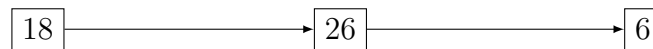
Zadanie polega na utworzeniu symulacji obsługi rezerwacji w kinie. Program umożliwia dodanie seansu, rezerwację i zakup biletów, wypisanie rezerwacji a także wypisywanie ich w zależności z uwzględnieniem daty oraz numeru sali.

## 2 Analiza zadania

Zagadnienie przedstawia problem quasi-bazy danych.

### 2.1 Struktury danych

W programie wykorzystano dwie listy jednokierunkowe. Pierwsza lista zawiera informacje o sali. Druga lista przechowuje informacje o rezerwacji tj. dacie, sali, bilecie i nazwie filmu. Taka struktura umożliwia sprawne zarządzanie rezerwacjami.

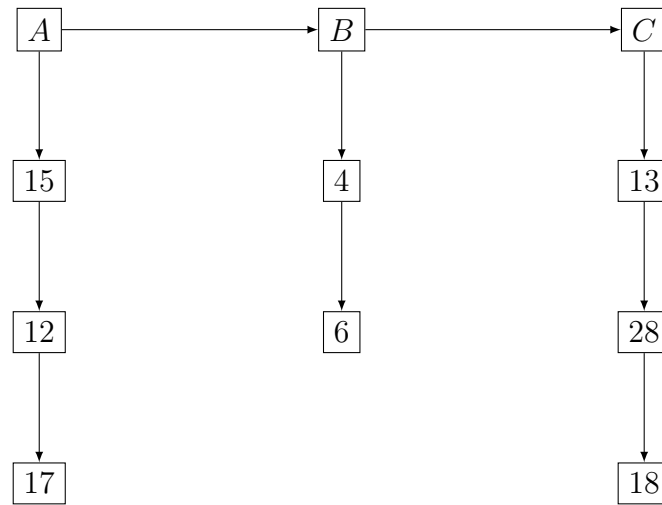


Rysunek 1: Przykład listy jednokierunkowej.

Uważam, że strukturę można usprawnić zmieniając ją w listę podwieszaną. W ten sposób każda sala będzie miała swoją listę rezerwacji, co znacząco usprawni działanie programu.

## 2.2 Algorytmy

---



Rysunek 2: Przykład listy podwieszanej.

## 2.2 Algorytmy

Program dodaje rezerwacje do listy jednokierunkowej uwzględniając, żeby nie utworzyć dwóch sal o tej samej dacie i jej unikalnym numerze. Wypisywanie list odbywa się poprzez iteracyjne przejście, z przeładowaną funkcją, uwzględniającą datę i nazwę filmu. [1]

## 3 Specyfikacja zewnętrzna

Program jest uruchamiany z konsoli. Przy uruchomieniu programu pojawia się menu:

```
Witaj w programie do obsługi rezerwacji kin
```

```
-----
```

Obsługiwane komendy:

instrukcja

zapisz <nazwapliku>

odczytaj <nazwapliku>

wyjdź

dodajFilm <data> <sala> <nazwaFilmu>

dodajFilm <data> <sala> <nazwaFilmu>

zabookuj <rzad> <kolumna> <data> <nazwaFilmu>

kup <rzad> <kolumna> <data> <nazwaFilmu>

odbookuj <rzad> <kolumna> <data> <nazwaFilmu>

### 3 Specyfikacja zewnętrzna

---

wypiszSale  
rezerwacje [<data>] [<nazwaFilmu>]

instrukcja

powoduje wyświetlenie instrukcji.

zapisz <nazwapliku>

powoduje zapis całego programu do pliku przekazanego przez argument `nazwapliku`.

odczytaj <nazwapliku>

powoduje odczyt z pliku przekazanego przez argument `nazwapliku`.

wyjdź

powoduje koniec programu.

dodajFilm <data> <sala> <nazwaFilmu>

powoduje dodanie nowego filmu z datą, salą i nazwą przekazaną przez argumenty `data`, `sala` i `nazwaFilmu`.

usunFilm <data> <sala> <nazwaFilmu>

powoduje usunięcie filmu z datą, salą i nazwą przekazaną przez argumenty `data`, `sala` i `nazwaFilmu`.

zabookuj <rzad> <kolumna> <data> <nazwaFilmu>

powoduje zarezerwowanie biletu na film o nazwie `nazwaFilmu` na fotelu przekazanym przez argumenty `rzad` i `kolumna` z datą `data`.

kup <rzad> <kolumna> <data> <nazwaFilmu>

powoduje kupienie biletu na film o nazwie `nazwaFilmu` na fotelu przekazanym przez argumenty `rzad` i `kolumna` z datą `data`.

```
odbookuj <rzad> <kolumna> <data> <nazwaFilmu>
```

powoduje usunięcie rezerwacji na film o nazwie `nazwaFilmu` na fotelu przekazanym przez argumenty `rzad` i `kolumna` z datą `data`.

```
wypiszSale
```

powoduje wypisanie na konsolę wszystkich sal filmowych.

```
rezerwacje [<data>] [<nazwaFilmu>]
```

powoduje wypisanie na konsolę wszystkich rezerwacji z opcjonalnymi parametrami `data` i `nazwaFilmu`.

### **3.1 Szczegółowy opis typów i funkcji**

Szczegółowy opis typów i funkcji zawarty jest w załączniku.

## **4 Testowanie**

Program został sprawdzony pod kątem wycieków pamięci przy użyciu biblioteki `nvwa` i `-fsanitize=leak` (biblioteki Clang).

## **5 Wnioski**

Program do obsługi rezerwacji nie był programem trudnym, ale kluczowe było zastosowanie odpowiednich struktur.

## **Literatura**

[1] Krzysztof Simiński. Wykłady z podstaw programowania komputerów.

Dodatek  
Szczegółowy opis typów  
i funkcji

Projekt zaliczeniowy z PK3-SSI

Wygenerowano przez Doxygen 1.8.20





<b>1 Indeks klas</b>	<b>1</b>
1.1 Lista klas	1
<b>2 Indeks plików</b>	<b>3</b>
2.1 Lista plików	3
<b>3 Dokumentacja klas</b>	<b>5</b>
3.1 Dokumentacja klasy Bilet	5
3.1.1 Opis szczegółowy	6
3.1.2 Dokumentacja konstruktora i destruktora	6
3.1.2.1 Bilet() [1/3]	6
3.1.2.2 Bilet() [2/3]	6
3.1.2.3 Bilet() [3/3]	6
3.1.3 Dokumentacja funkcji składowych	7
3.1.3.1 operator=()	7
3.1.3.2 ustawDate()	7
3.1.3.3 ustawID()	7
3.1.3.4 ustawKolumne()	8
3.1.3.5 ustawNazweFilmu()	8
3.1.3.6 ustawRzad()	8
3.1.3.7 ustawSale()	8
3.1.3.8 ustawStatus()	9
3.1.3.9 wypisz()	9
3.1.3.10 zwrocDate()	9
3.1.3.11 zwrocID()	9
3.1.3.12 zwrocKolumne()	10
3.1.3.13 zwrocNazweFilmu()	10
3.1.3.14 zwrocRzad()	10
3.1.3.15 zwrocSale()	10
3.1.3.16 zwrocStatus()	11
3.1.3.17 zwrocStatusJakoNapis()	11
3.1.4 Dokumentacja przyjaciół i funkcji związanych	11
3.1.4.1 operator<<	11
3.2 Dokumentacja klasy Data	11
3.2.1 Opis szczegółowy	12
3.2.2 Dokumentacja konstruktora i destruktora	12
3.2.2.1 Data() [1/2]	12
3.2.2.2 Data() [2/2]	12
3.2.2.3 ~Data()	13
3.2.3 Dokumentacja funkcji składowych	13
3.2.3.1 czyDatyKoliduja()	13
3.2.3.2 czyKoniecMiesiaca()	13
3.2.3.3 operator++()	14

3.2.3.4 operator=()	14
3.2.3.5 operator==()	14
3.2.3.6 operator>=()	15
3.2.3.7 ustawDzien()	15
3.2.3.8 ustawMiesiac()	15
3.2.3.9 ustawRok()	16
3.2.3.10 zwrocDzien()	16
3.2.3.11 zwrocMiesiac()	16
3.2.3.12 zwrocRok()	16
3.2.4 Dokumentacja przyjaciół i funkcji związanych	17
3.2.4.1 operator<<	17
3.2.4.2 operator>=	17
3.2.4.3 operator>>	17
3.3 Dokumentacja klasy ListaRezerwacji	18
3.3.1 Opis szczegółowy	18
3.3.2 Dokumentacja konstruktora i destruktora	18
3.3.2.1 ListaRezerwacji() [1/2]	19
3.3.2.2 ListaRezerwacji() [2/2]	19
3.3.2.3 ~ListaRezerwacji()	19
3.3.3 Dokumentacja funkcji składowych	19
3.3.3.1 dodajNaKoniec()	19
3.3.3.2 dodajNaKoniecListy()	19
3.3.3.3 ileWolnychKrsesel()	20
3.3.3.4 kupBilet()	20
3.3.3.5 usunListe()	21
3.3.3.6 usunRezerwacje() [1/2]	21
3.3.3.7 usunRezerwacje() [2/2]	21
3.3.3.8 wypisz()	21
3.3.3.9 wypiszWszystkieRezerwacje()	22
3.3.3.10 wypiszZarezerwowaneBilety()	22
3.3.3.11 zarezerwuj()	22
3.3.3.12 zwrocIloscRezerwacji()	23
3.3.3.13 zwrocRezerwacjeNumer()	23
3.3.4 Dokumentacja atrybutów składowych	23
3.3.4.1 pHead	23
3.4 Dokumentacja klasy ListaSal	23
3.4.1 Opis szczegółowy	24
3.4.2 Dokumentacja konstruktora i destruktora	24
3.4.2.1 ListaSal()	24
3.4.2.2 ~ListaSal()	24
3.4.3 Dokumentacja funkcji składowych	24
3.4.3.1 dodajNaKoniecListy()	24

3.4.3.2 dodajNaKoniecListyGut()	25
3.4.3.3 usunListe()	25
3.4.3.4 wypiszListe()	25
3.4.3.5 zwrocIloscSal()	25
3.4.3.6 zwrocSaleNumer()	25
3.4.3.7 zwrocTakaSamaSale()	26
3.4.4 Dokumentacja atrybutów składowych	26
3.4.4.1 pHead	26
3.5 Dokumentacja klasy Plik	26
3.5.1 Opis szczegółowy	27
3.5.2 Dokumentacja funkcji składowych	27
3.5.2.1 instrukcja()	27
3.5.2.2 odczytajZPliku()	27
3.5.2.3 zapiszDoPliku()	27
3.6 Dokumentacja klasy Rezerwacja	28
3.6.1 Opis szczegółowy	29
3.6.2 Dokumentacja konstruktora i destruktor	29
3.6.2.1 Rezerwacja() [1/2]	29
3.6.2.2 Rezerwacja() [2/2]	29
3.6.2.3 ~Rezerwacja()	29
3.6.3 Dokumentacja funkcji składowych	29
3.6.3.1 operator=()	29
3.6.3.2 ustawDate()	30
3.6.3.3 ustawLiczbeGosci()	30
3.6.3.4 ustawNazwe()	30
3.6.3.5 ustawSale()	30
3.6.3.6 ustawStatusNaFotelu()	31
3.6.3.7 wypisz()	31
3.6.3.8 zwrocBilet()	31
3.6.3.9 zwrocDate()	32
3.6.3.10 zwrocIleSprzedanych()	32
3.6.3.11 zwrocIleSprzedanychIZarezerwowanych()	32
3.6.3.12 zwrocIloscWizyt()	32
3.6.3.13 zwrocNazwe()	32
3.6.3.14 zwrocSale()	33
3.6.4 Dokumentacja przyjaciół i funkcji związanych	33
3.6.4.1 operator<<	33
3.6.5 Dokumentacja atrybutów składowych	33
3.6.5.1 pNext	33
3.7 Dokumentacja klasy Sala	33
3.7.1 Opis szczegółowy	34
3.7.2 Dokumentacja konstruktora i destruktor	34

3.7.2.1 Sala() [1/4]	34
3.7.2.2 Sala() [2/4]	34
3.7.2.3 Sala() [3/4]	35
3.7.2.4 Sala() [4/4]	35
3.7.3 Dokumentacja funkcji składowych	35
3.7.3.1 operator=() [1/2]	35
3.7.3.2 operator=() [2/2]	36
3.7.3.3 ustawKolumne()	36
3.7.3.4 ustawNumer()	36
3.7.3.5 ustawRzad()	37
3.7.3.6 zwrocKolumne()	37
3.7.3.7 zwrocNumer()	37
3.7.3.8 zwrocPojemonsc()	38
3.7.3.9 zwrocRzad()	38
3.7.4 Dokumentacja przyjaciół i funkcji związanych	38
3.7.4.1 operator<<	38
3.7.5 Dokumentacja atrybutów składowych	38
3.7.5.1 pNext	38
<b>4 Dokumentacja plików</b>	<b>41</b>
4.1 Dokumentacja pliku kod/Bilet.hpp	41
4.2 Dokumentacja pliku kod/Data.hpp	41
4.3 Dokumentacja pliku kod/Plik.hpp	41
4.3.1 Dokumentacja funkcji	42
4.3.1.1 odczytajZKonsoli()	42
4.4 Dokumentacja pliku kod/Rezerwacja.hpp	42
4.5 Dokumentacja pliku kod/Rezerwacje.hpp	42
4.6 Dokumentacja pliku kod/Sala.hpp	43
4.7 Dokumentacja pliku kod/Status.h	43
4.7.1 Dokumentacja typów wyliczanych	43
4.7.1.1 Status	43
<b>Indeks</b>	<b>45</b>

# Rozdział 1

## Indeks klas

### 1.1 Lista klas

Tutaj znajdują się klasy, struktury, unie i interfejsy wraz z ich krótkimi opisami:

<a href="#">Bilet</a>	5
<a href="#">Data</a>	11
<a href="#">ListaRezerwacji</a>	18
<a href="#">ListaSal</a>	23
<a href="#">Plik</a>	26
<a href="#">Rezerwacja</a>	28
<a href="#">Sala</a>	33



## Rozdział 2

# Indeks plików

### 2.1 Lista plików

Tutaj znajduje się lista wszystkich plików z ich krótkimi opisami:

<a href="#">kod/Bilet.hpp</a>	41
<a href="#">kod/Data.hpp</a>	41
<a href="#">kod/Plik.hpp</a>	41
<a href="#">kod/Rezerwacja.hpp</a>	42
<a href="#">kod/Rezerwacje.hpp</a>	42
<a href="#">kod/Sala.hpp</a>	43
<a href="#">kod/Status.h</a>	43





## Rozdział 3

# Dokumentacja klas

### 3.1 Dokumentacja klasy Bilet

```
#include <Bilet.hpp>
```

Diagram współpracy dla Bilet:

#### Metody publiczne

- `Bilet ()`
- `Bilet (const Sala *_sala, const std::string &_nazwa, const Data &_data, int _rzad, int _kolumna, const Status &_status)`
- `Bilet (const Sala *_sala)`
- `Bilet & operator= (const Bilet &obj)`
- `std::string zwrocID () const`
- `std::string zwrocNazweFilmu () const`
- `const Sala * zwrocSale () const`
- `Data zwrocDate () const`
- `const int zwrocRzad () const`
- `const int zwrocKolumne () const`
- `Status zwrocStatus () const`
- `std::string zwrocStatusJakoNapis () const`
- `void ustawID (const std::string _ID)`
- `void ustawNazweFilmu (const std::string _nazwa)`
- `void ustawSale (const Sala *_sala)`
- `void ustawDate (const Data &_data)`
- `void ustawRzad (int _rzad)`
- `void ustawKolumne (int _kolumna)`
- `void ustawStatus (const Status &_status)`
- `void wypisz () const`

#### Przyjaciele

- `std::ostream & operator<< (std::ostream &os, const Bilet &bilet)`

### 3.1.1 Opis szczegółowy

Klasa reprezentująca bilet tj. unikalne ID, nazwa zakupionego filmu, sala, data, krzeselko i status biletu

### 3.1.2 Dokumentacja konstruktora i destruktora

#### 3.1.2.1 Bilet() [1/3]

```
Bilet::Bilet ( )
```

Konstruktor bezargumentowy- lista inicjalizacyjna

#### 3.1.2.2 Bilet() [2/3]

```
Bilet::Bilet (
    const Sala * _sala,
    const std::string & _nazwa,
    const Data & _data,
    int _rzad,
    int _kolumna,
    const Status & _status )
```

Konstruktor wieloargumentowy- lista inicjalizacyjna

##### Parametry

<code>_sala</code>	sala reprezentowana w klasie
<code>_nazwa</code>	nazwa filmu -  -
<code>_data</code>	data -  -
<code>_rzad</code>	rząd krzeselka -  -
<code>_kolumna</code>	kolumna krzeselka -  -
<code>_status</code>	status biletu -  -

#### 3.1.2.3 Bilet() [3/3]

```
Bilet::Bilet (
    const Sala * sala )
```

Konstruktor jednoargumentowy- lista inicjalizacyjna

##### Parametry

<code>sala</code>	sala reprezentowana w klasie
-------------------	------------------------------

### 3.1.3 Dokumentacja funkcji składowych

#### 3.1.3.1 operator=()

```
Bilet& Bilet::operator= (
    const Bilet & obj )
```

Przeładowany operator przypisania

Parametry

<i>obj</i>	sala do przypisania
------------	---------------------

Zwraca

zwraca przypisaną salę

#### 3.1.3.2 ustawDate()

```
void Bilet::ustawDate (
    const Data & _data )
```

Metoda typu setter, ustawiająca datę

Parametry

<i>_data</i>	data do ustawienia
--------------	--------------------

#### 3.1.3.3 ustawID()

```
void Bilet::ustawID (
    const std::string _ID )
```

Metoda typu setter, ustawiająca ID

Parametry

<i>_ID</i>	ID do ustawienia
------------	------------------

#### 3.1.3.4 ustawKolumne()

```
void Bilet::ustawKolumne (
    int _kolumna )
```

Metoda typu setter, ustawiająca kolumnę krzesła

Parametry

<code>_kolumna</code>	kolumna do ustawienia
-----------------------	-----------------------

#### 3.1.3.5 ustawNazweFilmu()

```
void Bilet::ustawNazweFilmu (
    const std::string _nazwa )
```

Metoda typu setter, ustawiająca nazwę filmu

Parametry

<code>_nazwa</code>	nazwa do ustawienia
---------------------	---------------------

#### 3.1.3.6 ustawRzad()

```
void Bilet::ustawRzad (
    int _rzad )
```

Metoda typu setter, ustawiająca rząd krzesła

Parametry

<code>_rzad</code>	rząd do ustawienia
--------------------	--------------------

#### 3.1.3.7 ustawSale()

```
void Bilet::ustawSale (
    const Sala * _sala )
```

Metoda typu setter, ustawiająca salę

## Parametry

<code>_sala</code>	sala do ustawienia
--------------------	--------------------

**3.1.3.8 ustawStatus()**

```
void Bilet::ustawStatus (
    const Status & _status )
```

Metoda typu setter, ustawiająca status biletu

## Parametry

<code>_status</code>	status do ustawienia
----------------------	----------------------

**3.1.3.9 wypisz()**

```
void Bilet::wypisz ( ) const
```

**3.1.3.10 zwrocDate()**

```
Data Bilet::zwrocDate ( ) const
```

Metoda typu getter, zwracająca datę

## Zwraca

data

**3.1.3.11 zwrocID()**

```
std::string Bilet::zwrocID ( ) const
```

Metoda typu getter, zwracająca unikalne ID biletu

## Zwraca

numer ID

#### 3.1.3.12 zwrocKolumne()

```
const int Bilet::zwrocKolumne ( ) const
```

Metoda typu getter, zwracająca kolumnę krzesła

**Zwraca**

kolumna

#### 3.1.3.13 zwrocNazweFilmu()

```
std::string Bilet::zwrocNazweFilmu ( ) const
```

Metoda typu getter, zwracająca nazwę filmu

**Zwraca**

nazwa filmu

#### 3.1.3.14 zwrocRzad()

```
const int Bilet::zwrocRzad ( ) const
```

Metoda typu getter, zwracająca rząd krzesła

**Zwraca**

rząd

#### 3.1.3.15 zwrocSale()

```
const Sala* Bilet::zwrocSale ( ) const
```

Metoda typu getter, zwracająca salę

**Zwraca**

sala

### 3.1.3.16 zwrocStatus()

```
Status Bilet::zwrocStatus ( ) const
```

Metoda typu getter, zwracająca status

Zwraca

status

### 3.1.3.17 zwrocStatusJakoNapis()

```
std::string Bilet::zwrocStatusJakoNapis ( ) const
```

Metoda zwracająca enum jako string

Zwraca

status jako napis

## 3.1.4 Dokumentacja przyjaciół i funkcji związanych

### 3.1.4.1 operator<<

```
std::ostream& operator<< (
    std::ostream & os,
    const Bilet & bilet ) [friend]
```

Zaprzyjaźniony operator strumieniowy służący do wypisania biletu do strumienia

Parametry

<i>os</i>	output stream
<i>bilet</i>	bilet do wypisania @retrun zwraca strumień

Dokumentacja dla tej klasy została wygenerowana z pliku:

- kod/[Bilet.hpp](#)

## 3.2 Dokumentacja klasy Data

```
#include <Data.hpp>
```

Diagram współpracy dla Data:

## Metody publiczne

- `Data ()`
- `Data (int _dzien, int _miesiac, int _rok)`
- `Data & operator= (const Data &obj)`
- `bool operator== (const Data &obj)`
- `bool operator>= (const Data &obj)`
- `Data & operator++ ()`
- `~Data ()`
- `int zwrocDzien () const`
- `int zwrocMiesiac () const`
- `int zwrocRok () const`
- `void ustawDzien (const int _dzien)`
- `void ustawMiesiac (const int _miesiac)`
- `void ustawRok (const int _rok)`
- `bool czyKoniecMiesiaca () const`
- `bool czyDatyKoliduja (const Data pocz1, const Data koniec1, const Data pocz2, const Data koniec2)`

## Przyjaciele

- `bool operator>= (const Data &pierwsza, const Data &druga)`
- `std::ostream & operator<< (std::ostream &os, const Data &data)`
- `std::istream & operator>> (std::istream &is, Data &data)`

### 3.2.1 Opis szczegółowy

Klasa reprezentująca date tj. dzień, miesiąc i rok

### 3.2.2 Dokumentacja konstruktora i destruktor

#### 3.2.2.1 `Data()` [1/2]

```
Data::Data ( )
```

Konstruktor bezargumentowy- lista inicjalizacyjna

#### 3.2.2.2 `Data()` [2/2]

```
Data::Data (
    int _dzien,
    int _miesiac,
    int _rok )
```

Konstruktor wieloargumentowy- lista inicjalizacyjna



## Parametry

<code>_dzien</code>	dzień reprezentowany w klasie
<code>_miesiac</code>	miesiąc -  -
<code>_rok</code>	rok -  -

## 3.2.2.3 ~Data()

```
Data::~~Data ( )
```

Destruktor

## 3.2.3 Dokumentacja funkcji składowych

## 3.2.3.1 czyDatyKoliduja()

```
bool Data::czyDatyKoliduja (
    const Data pocz1,
    const Data koniec1,
    const Data pocz2,
    const Data koniec2 )
```

Metoda sprawdzająca czy dwa przedziały dat kolidują ze sobą

## Parametry

<code>pocz1</code>	początek pierwszego przedziału
<code>koniec1</code>	koniec pierwszego przedziału
<code>pocz2</code>	początek drugiego przedziału
<code>koniec2</code>	koniec drugiego przedziału

## Zwraca

true jeżeli daty kolidują

## 3.2.3.2 czyKoniecMiesiaca()

```
bool Data::czyKoniecMiesiaca ( ) const
```

Metoda sprawdzająca czy jest koniec miesiąca

**Zwraca**

true jeżeli dzień w miesiącu jest poprawny

**3.2.3.3 operator++()**

```
Data& Data::operator++ ( )
```

Przeładowany operator inkrementacji, inkrementuje datę uwzględniając warunek 12 miesięcy w roku kalendarzowym

**Zwraca**

metoda zwraca obiekt

**3.2.3.4 operator=()**

```
Data& Data::operator= (
    const Data & obj )
```

Przeładowany operator przypisania

**Parametry**

<i>obj</i>	data do przypisania
------------	---------------------

**Zwraca**

zwraca przypisaną datę

**3.2.3.5 operator==()**

```
bool Data::operator== (
    const Data & obj )
```

Przeładowany operator porównania, porównujący dwie daty

**Parametry**

<i>obj</i>	data do porównania
------------	--------------------

**Zwraca**

metoda zwraca true jeżeli daty są takie same

**3.2.3.6 operator>=()**

```
bool Data::operator>= (
    const Data & obj )
```

Przeładowany operator >=, porównujący dwie daty

**Parametry**

<i>obj</i>	referencja do obiektu który jest typu <a href="#">Data</a>
------------	--

**Zwraca**

metoda zwraca true jeżeli this->data jest większa

**3.2.3.7 ustawDzien()**

```
void Data::ustawDzien (
    const int _dzien )
```

Metoda typu setter, ustawiająca dzień

**Parametry**

<i>_dzien</i>	dzień do ustawienia
---------------	---------------------

**3.2.3.8 ustawMiesiac()**

```
void Data::ustawMiesiac (
    const int _miesiac )
```

Metoda typu setter, ustawiająca miesiąc

**Parametry**

<i>_miesiac</i>	miesiąc do ustawienia
-----------------	-----------------------

### 3.2.3.9 ustawRok()

```
void Data::ustawRok (
    const int _rok )
```

Metoda typu setter, ustawiająca rok

Parametry

<code>_rok</code>	rok do ustawienia
-------------------	-------------------

### 3.2.3.10 zwrocDzien()

```
int Data::zwrocDzien ( ) const
```

Metoda typu getter, zwracająca dzień

Zwraca

dzień

### 3.2.3.11 zwrocMiesiac()

```
int Data::zwrocMiesiac ( ) const
```

Metoda typu getter, zwracająca miesiąc

Zwraca

miesiąc

### 3.2.3.12 zwrocRok()

```
int Data::zwrocRok ( ) const
```

Metoda typu getter, zwracająca rok

Zwraca

rok

## 3.2.4 Dokumentacja przyjaciół i funkcji związanych

### 3.2.4.1 operator<<

```
std::ostream& operator<< (
    std::ostream & os,
    const Data & data ) [friend]
```

Zaprzyjaźniony operator strumieniowy służący do wypisania daty do strumienia

#### Parametry

<i>os</i>	output stream
<i>data</i>	data do wypisania @retrun zwraca strumień

### 3.2.4.2 operator>=

```
bool operator>= (
    const Data & pierwsza,
    const Data & druga ) [friend]
```

Zaprzyjaźniony operator większy lub równy, porównuje dwie daty

#### Parametry

<i>pierwsza</i>	pierwsza data do porównania
<i>druga</i>	druga data do porównania

#### Zwraca

true jeżeli pierwsza data jest większa

### 3.2.4.3 operator>>

```
std::istream& operator>> (
    std::istream & is,
    Data & data ) [friend]
```

Zaprzyjaźniony operator strumieniowy służący do wpisania daty do strumienia

## Parametry

<i>is</i>	input stream
<i>data</i>	data do wpisania @retrun zwraca strumień

Dokumentacja dla tej klasy została wygenerowana z pliku:

- kod/[Data.hpp](#)

### 3.3 Dokumentacja klasy ListaRezerwacji

```
#include <Rezerwacje.hpp>
```

Diagram współpracy dla ListaRezerwacji:

#### Metody publiczne

- [ListaRezerwacji](#) ()
- [ListaRezerwacji](#) (const [ListaRezerwacji](#) &obj)
- [~ListaRezerwacji](#) ()
- void [dodajNaKoniec](#) ([Rezerwacja](#) \*rezerwacja)
- void [dodajNaKoniecListy](#) ([Sala](#) \*sala, const [Data](#) &data, const std::string &nazwa)
- void [wypisz](#) ()
- int [ileWolnychKrsesel](#) (const [Data](#) &data, const std::string &nazwa)
- void [wypiszZarezerwowaneBilety](#) (const [Data](#) &data, const std::string &nazwa)
- void [wypiszWszystkieRezerwacje](#) (const [Data](#) &data)
- int [zwrocIloscRezerwacji](#) ()
- [Rezerwacja](#) \* [zwrocRezerwacjeNumer](#) (const int &index)
- void [usunRezerwacje](#) (const std::string &nazwa)
- void [zarezerwuj](#) (const int rzad, const int kolumna, const [Data](#) &data, const std::string nazwa)
- void [usunRezerwacje](#) (const int rzad, const int kolumna, const [Data](#) &data, const std::string nazwa)
- void [kupBilet](#) (const int rzad, const int kolumna, const [Data](#) &data, const std::string nazwa)
- void [usunListe](#) ()

#### Atrybuty publiczne

- [Rezerwacja](#) \* [pHead](#)  
*wskaźnik na głowę listy*

#### 3.3.1 Opis szczegółowy

Lista jednokierunkowa reprezentująca rezerwacje

#### 3.3.2 Dokumentacja konstruktora i destruktor

### 3.3.2.1 ListaRezerwacji() [1/2]

```
ListaRezerwacji::ListaRezerwacji ( ) [inline]
```

Konstruktor bezargumentowy- lista inicjalizacyjna. Inicjalizuje głowę do nullptr.

### 3.3.2.2 ListaRezerwacji() [2/2]

```
ListaRezerwacji::ListaRezerwacji (
    const ListaRezerwacji & obj )
```

Przeładowany konstruktor kopiujący

Parametry

<i>obj</i>	lista rezerwacji do skopiowania
------------	---------------------------------

### 3.3.2.3 ~ListaRezerwacji()

```
ListaRezerwacji::~~ListaRezerwacji ( )
```

Destruktor zwalniający pamięć listy

## 3.3.3 Dokumentacja funkcji składowych

### 3.3.3.1 dodajNaKoniec()

```
void ListaRezerwacji::dodajNaKoniec (
    Rezerwacja * rezerwacja )
```

Metoda dodająca rezerwacje na początek listy jednokierunkowej

Parametry

<i>rezerwacja</i>	rezerwacja do dodania
-------------------	-----------------------

### 3.3.3.2 dodajNaKoniecListy()

```
void ListaRezerwacji::dodajNaKoniecListy (
    Sala * sala,
```

```
const Data & data,
const std::string & nazwa )
```

Metoda dodająca rezerwacje na początek listy jednokierunkowej z uwzględnieniem braku powtórzeń

#### Parametry

<i>sala</i>	sala do dodania
<i>data</i>	data do dodania
<i>nazwa</i>	nazwa filmu

#### 3.3.3.3 ileWolnychKrzeseł()

```
int ListaRezerwacji::ileWolnychKrzeseł (
    const Data & data,
    const std::string & nazwa )
```

Metoda zwracająca ilość wolnych krzeseł dla danej rezerwacji

#### Parametry

<i>data</i>	data rezerwacji
<i>nazwa</i>	nazwa filmu

#### Zwraca

ile wolnych krzeseł

#### 3.3.3.4 kupBilet()

```
void ListaRezerwacji::kupBilet (
    const int rzad,
    const int kolumna,
    const Data & data,
    const std::string nazwa )
```

Metoda kupująca bilet dla konkretnego filmu

#### Parametry

<i>rzad</i>	rząd krzesła
<i>kolumna</i>	kolumna krzesła
<i>data</i>	data filmu
<i>nazwa</i>	nazwa filmu



#### 3.3.3.5 usunListe()

```
void ListaRezerwacji::usunListe ( )
```

Metoda usuwająca całą listę jednokierunkową

#### 3.3.3.6 usunRezerwacje() [1/2]

```
void ListaRezerwacji::usunRezerwacje (
    const int rzad,
    const int kolumna,
    const Data & data,
    const std::string nazwa )
```

Metoda usuwająca rezerwację dla konkretnego filmu

##### Parametry

<i>rzad</i>	rząd krzesła
<i>kolumna</i>	kolumna krzesła
<i>data</i>	data filmu
<i>nazwa</i>	nazwa filmu

#### 3.3.3.7 usunRezerwacje() [2/2]

```
void ListaRezerwacji::usunRezerwacje (
    const std::string & nazwa )
```

Metoda usuwająca element listy o podanej nazwie

##### Parametry

<i>nazwa</i>	nazwa do usunięcia
--------------	--------------------

#### 3.3.3.8 wypisz()

```
void ListaRezerwacji::wypisz ( )
```

Metoda wypisująca listę jednokierunkową

### 3.3.3.9 wypiszWszystkieRezerwacje()

```
void ListaRezerwacji::wypiszWszystkieRezerwacje (
    const Data & data )
```

Metoda wypisująca zarezerwowane bilety dla rezerwacji danego dnia

Parametry

<i>data</i>	data
-------------	------

### 3.3.3.10 wypiszZarezerwowaneBilety()

```
void ListaRezerwacji::wypiszZarezerwowaneBilety (
    const Data & data,
    const std::string & nazwa )
```

Metoda wypisująca zarezerwowane bilety dla rezerwacji danego dnia i o podanej nazwie

Parametry

<i>data</i>	data
<i>nazwa</i>	nazwa filmu

### 3.3.3.11 zarezerwuj()

```
void ListaRezerwacji::zarezerwuj (
    const int rzad,
    const int kolumna,
    const Data & data,
    const std::string nazwa )
```

Metoda rezerwująca bilet dla konkretnego filmu

Parametry

<i>rzad</i>	rząd krzesła
<i>kolumna</i>	kolumna krzesła
<i>data</i>	data filmu
<i>nazwa</i>	nazwa filmu

### 3.3.3.12 zwrocIloscRezerwacji()

```
int ListaRezerwacji::zwrocIloscRezerwacji ( )
```

Metoda zwracająca ilość wszystkich elementów listy jednokierunkowej

**Zwraca**

ile elementów listy

### 3.3.3.13 zwrocRezerwacjeNumer()

```
Rezerwacja* ListaRezerwacji::zwrocRezerwacjeNumer (
    const int & index )
```

Metoda zwracająca element listy o podanym indeksie

**Parametry**

<i>index</i>	indeks do iteracji
--------------	--------------------

**Zwraca**

rezerwacja o indeksie

## 3.3.4 Dokumentacja atrybutów składowych

### 3.3.4.1 pHead

```
Rezerwacja* ListaRezerwacji::pHead
```

wskaźnik na głowę listy

Dokumentacja dla tej klasy została wygenerowana z pliku:

- kod/[Rezerwacje.hpp](#)

## 3.4 Dokumentacja klasy ListaSal

```
#include <Rezerwacje.hpp>
```

Diagram współpracy dla ListaSal:

## Metody publiczne

- `ListaSal ()`
- `~ListaSal ()`
- `void dodajNaKoniecListy (Sala *sala)`
- `void dodajNaKoniecListyGut (Sala *sala)`
- `Sala * zwrocTakaSamaSale (Sala *sala)`
- `void wypiszListe ()`
- `Sala * zwrocSaleNumer (const int &index)`
- `int zwrocIloscSal ()`
- `void usunListe ()`

## Atrybuty publiczne

- `Sala * pHead`  
*wskaźnik na głowę listy*

### 3.4.1 Opis szczegółowy

Lista jednokierunkowa reprezentująca sale

### 3.4.2 Dokumentacja konstruktora i destruktora

#### 3.4.2.1 ListaSal()

```
ListaSal::ListaSal ( ) [inline]
```

Konstruktor bezargumentowy- lista inicjalizacyjna. Inicjalizuje głowę do nullptr.

#### 3.4.2.2 ~ListaSal()

```
ListaSal::~~ListaSal ( )
```

Destruktor zwalniający pamięć listy

### 3.4.3 Dokumentacja funkcji składowych

#### 3.4.3.1 dodajNaKoniecListy()

```
void ListaSal::dodajNaKoniecListy (  
    Sala * sala )
```

Metoda dodająca rezerwacje na początek listy jednokierunkowej

## Parametry

<i>sala</i>	sala do dodania
-------------	-----------------

**3.4.3.2 dodajNaKoniecListyGut()**

```
void ListaSal::dodajNaKoniecListyGut (
    Sala * sala )
```

Metoda dodająca salę na początek listy jednokierunkowej z uwzględnieniem braku powtórzeń

## Parametry

<i>sala</i>	sala do dodania
-------------	-----------------

**3.4.3.3 usunListe()**

```
void ListaSal::usunListe ( )
```

Metoda usuwająca całą listę jednokierunkową

**3.4.3.4 wypiszListe()**

```
void ListaSal::wypiszListe ( )
```

Metoda wypisująca listę jednokierunkowa

**3.4.3.5 zwrocIloscSal()**

```
int ListaSal::zwrocIloscSal ( )
```

Metoda zwracająca ilość wszystkich elementów listy jednokierunkowej

## Zwraca

ile elementów listy

**3.4.3.6 zwrocSaleNumer()**

```
Sala* ListaSal::zwrocSaleNumer (
    const int & index )
```

Metoda zwracająca element listy o podanym indeksie

**Parametry**

<i>index</i>	indeks do iteracji
--------------	--------------------

**Zwraca**

wskaźnik na salę o indeksie

**3.4.3.7 zwrocTakaSamaSale()**

```
Sala* ListaSal::zwrocTakaSamaSale (
    Sala * sala )
```

Metoda wyszukająca w liście taki sam element

**Parametry**

<i>sala</i>	sala do znalezienia
-------------	---------------------

**Zwraca**

wskaźnik na salę lub nullptr

**3.4.4 Dokumentacja atrybutów składowych****3.4.4.1 pHead**

```
Sala* ListaSal::pHead
```

wskaźnik na głowę listy

Dokumentacja dla tej klasy została wygenerowana z pliku:

- kod/[Rezerwacje.hpp](#)

**3.5 Dokumentacja klasy Plik**

```
#include <Plik.hpp>
```

Diagram współpracy dla Plik:

## Metody publiczne

- void `zapiszDoPliku` (std::string &nazwaPliku, `ListaSal` \*pGlowaSal, `ListaRezerwacji` \*pGlowaRezerwacji)
- void `odczytajZPliku` (std::string &nazwaPliku, `ListaSal` \*pGlowaSal, `ListaRezerwacji` \*pGlowaRezerwacji)
- void `instrukcja` ()

### 3.5.1 Opis szczegółowy

Klasa pomocnicza służąca do obsługi plików i menu

### 3.5.2 Dokumentacja funkcji składowych

#### 3.5.2.1 instrukcja()

```
void Plik::instrukcja ( )
```

Metoda wypisująca instrukcję obsługi

#### 3.5.2.2 odczytajZPliku()

```
void Plik::odczytajZPliku (
    std::string & nazwaPliku,
    ListaSal * pGlowaSal,
    ListaRezerwacji * pGlowaRezerwacji )
```

Metoda odczytująca wszystkie rezerwacje z pliku

Parametry

<i>nazwaPliku</i>	nazwapliku
<i>pGlowaSal</i>	głowa listy sal
<i>pGlowaRezerwacji</i>	głowa listy rezerwacji

#### 3.5.2.3 zapiszDoPliku()

```
void Plik::zapiszDoPliku (
    std::string & nazwaPliku,
    ListaSal * pGlowaSal,
    ListaRezerwacji * pGlowaRezerwacji )
```

Metoda zapisująca wszystkie rezerwacje do pliku

## Parametry

<i>nazwaPliku</i>	nazwapliku
<i>pGlowaSal</i>	głowa listy sal
<i>pGlowaRezerwacji</i>	głowa listy rezerwacji

Dokumentacja dla tej klasy została wygenerowana z pliku:

- kod/[Plik.hpp](#)

### 3.6 Dokumentacja klasy Rezerwacja

```
#include <Rezerwacja.hpp>
```

Diagram współpracy dla Rezerwacja:

#### Metody publiczne

- [Rezerwacja](#) ()
- [Rezerwacja](#) (const std::string &\_nazwa, const [Data](#) &data, const [Sala](#) \*sala)
- [Rezerwacja](#) & [operator=](#) (const [Rezerwacja](#) &obj)
- [~Rezerwacja](#) ()
- std::string [zwrocNazwe](#) () const
- [Data](#) [zwrocDate](#) () const
- int [zwrocIleSprzedanych](#) () const
- int [zwrocIleSprzedanychIZarezerwowanych](#) () const
- [Bilet](#) \* [zwrocBilet](#) (int \_rzad, int \_kolumna) const
- const [Sala](#) \* [zwrocSale](#) () const
- int [zwrocIloscWizyt](#) () const
- void [ustawNazwe](#) (std::string \_nazwa)
- void [ustawDate](#) (const [Data](#) &\_data)
- void [ustawStatusNaFotelu](#) (const int &\_rzad, const int &\_kolumna, const [Status](#) &\_status)
- void [ustawSale](#) (const [Sala](#) \*sala)
- void [ustawLiczbeGosci](#) (const int &\_liczba)
- void [wypisz](#) () const

#### Atrybuty publiczne

- [Rezerwacja](#) \* [pNext](#)  
*wskaźnik na następny element rezerwacji*

#### Przyjaciele

- std::ostream & [operator<<](#) (std::ostream &os, const [Rezerwacja](#) &rez)



### 3.6.1 Opis szczegółowy

Klasa reprezentująca rezerwację tj. nazwa filmu, data, sala, tablica dwuwymiarowa biletów

### 3.6.2 Dokumentacja konstruktora i destruktora

#### 3.6.2.1 Rezerwacja() [1/2]

```
Rezerwacja::Rezerwacja ( )
```

Konstruktor bezargumentowy- lista inicjalizacyjna

#### 3.6.2.2 Rezerwacja() [2/2]

```
Rezerwacja::Rezerwacja (
    const std::string & _nazwa,
    const Data & data,
    const Sala * sala )
```

Konstruktor wieloargumentowy- lista inicjalizacyjna

Parametry

<i>_nazwa</i>	nazwa filmu reprezentowana w klasie
<i>data</i>	data -  -
<i>sala</i>	sala -  -

#### 3.6.2.3 ~Rezerwacja()

```
Rezerwacja::~~Rezerwacja ( )
```

Destruktor czyszczący wektor

### 3.6.3 Dokumentacja funkcji składowych

#### 3.6.3.1 operator=()

```
Rezerwacja& Rezerwacja::operator= (
    const Rezerwacja & obj )
```

Przeładowany operator przypisania

**Parametry**

<i>obj</i>	rezerwacja do przypisania
------------	---------------------------

**Zwraca**

zwraca przypisaną salę

**3.6.3.2 ustawDate()**

```
void Rezerwacja::ustawDate (
    const Data & _data )
```

Metoda typu setter, ustawiająca datę

**Parametry**

<i>_data</i>	data do ustawienia
--------------	--------------------

**3.6.3.3 ustawLiczbeGosci()**

```
void Rezerwacja::ustawLiczbeGosci (
    const int & _liczba )
```

**3.6.3.4 ustawNazwe()**

```
void Rezerwacja::ustawNazwe (
    std::string _nazwa )
```

Metoda typu setter, ustawiająca nazwę filmu

**Parametry**

<i>_nazwa</i>	nazwa do ustawienia
---------------	---------------------

**3.6.3.5 ustawSale()**

```
void Rezerwacja::ustawSale (
    const Sala * sala )
```

Metoda typu setter, ustawiająca salę

Parametry

<i>sala</i>	sala do ustawienia
-------------	--------------------

### 3.6.3.6 ustawStatusNaFotelu()

```
void Rezerwacja::ustawStatusNaFotelu (
    const int & _rzad,
    const int & _kolumna,
    const Status & _status )
```

Metoda typu setter, ustawiająca status na fotelu

Parametry

<i>_rzad</i>	rząd krzeselka
<i>_kolumna</i>	kolumna krzeselka
<i>_status</i>	status do ustawienia

### 3.6.3.7 wypisz()

```
void Rezerwacja::wypisz ( ) const
```

### 3.6.3.8 zwrocBilet()

```
Bilet* Rezerwacja::zwrocBilet (
    int _rzad,
    int _kolumna ) const
```

Metoda typu getter, zwracająca bilet na danym krzeselku

Parametry

<i>_rzad</i>	rząd
<i>_kolumna</i>	kolumna

Zwraca

bilet

### 3.6.3.9 zwrocDate()

```
Data Rezerwacja::zwrocDate ( ) const
```

Metoda typu getter, zwracająca datę

**Zwraca**

data

### 3.6.3.10 zwrocIleSprzedanych()

```
int Rezerwacja::zwrocIleSprzedanych ( ) const
```

Metoda zwracająca ilość sprzedanych biletów dla danego seansu

**Zwraca**

ilość sprzedanych biletów

### 3.6.3.11 zwrocIleSprzedanychIZarezerwowanych()

```
int Rezerwacja::zwrocIleSprzedanychIZarezerwowanych ( ) const
```

Metoda zwracająca ilość sprzedanych i zarezerwowanych biletów dla danego seansu

**Zwraca**

ilość sprzedanych i zarezerwowanych biletów

### 3.6.3.12 zwrocIloscWizyt()

```
int Rezerwacja::zwrocIloscWizyt ( ) const
```

### 3.6.3.13 zwrocNazwe()

```
std::string Rezerwacja::zwrocNazwe ( ) const
```

Metoda typu getter, zwracająca nazwę filmu

**Zwraca**

nazwa filmu

#### 3.6.3.14 zwrocSale()

```
const Sala* Rezerwacja::zwrocSale ( ) const
```

Metoda typu getter, zwracająca salę

Zwraca

sala

### 3.6.4 Dokumentacja przyjaciół i funkcji związanych

#### 3.6.4.1 operator<<

```
std::ostream& operator<< (
    std::ostream & os,
    const Rezerwacja & rez ) [friend]
```

Zaprzyjaźniony operator strumieniowy służący do wypisania rezerwacji do strumienia

Parametry

<i>os</i>	output stream
<i>rez</i>	rezerwacja do wypisania @retrun zwraca strumień

### 3.6.5 Dokumentacja atrybutów składowych

#### 3.6.5.1 pNext

```
Rezerwacja* Rezerwacja::pNext
```

wskaźnik na następny element rezerwacji

Dokumentacja dla tej klasy została wygenerowana z pliku:

- kod/[Rezerwacja.hpp](#)

## 3.7 Dokumentacja klasy Sala

```
#include <Sala.hpp>
```

Diagram współpracy dla Sala:

## Metody publiczne

- [Sala](#) ()
- [Sala](#) (int \_numer, int \_rzad, int \_kolumna)
- [Sala](#) (int \_numer)
- [Sala](#) (const [Sala](#) &obj)
- [Sala](#) & [operator=](#) (const [Sala](#) &obj)
- [Sala](#) & [operator=](#) (const [Sala](#) &&obj)
- int [zwrocNumer](#) () const
- int [zwrocRzad](#) () const
- int [zwrocKolumne](#) () const
- int [zwrocPojemosc](#) () const
- void [ustawNumer](#) (int \_numer)
- void [ustawRzad](#) (int \_rzad)
- void [ustawKolumne](#) (int \_kolumna)

## Atrybuty publiczne

- [Sala](#) \* [pNext](#)  
*wskaźnik na następny element sali*

## Przyjaciele

- std::ostream & [operator<<](#) (std::ostream &os, const [Sala](#) &sala)

### 3.7.1 Opis szczegółowy

Klasa reprezentująca salę tj. ilość rzędów, kolumn i jej numer

### 3.7.2 Dokumentacja konstruktora i destruktor

#### 3.7.2.1 [Sala\(\)](#) [1/4]

```
Sala::Sala ( )
```

Konstruktor bezargumentowy- lista inicjalizacyjna

#### 3.7.2.2 [Sala\(\)](#) [2/4]

```
Sala::Sala (
    int _numer,
    int _rzad,
    int _kolumna )
```

Konstruktor wieloargumentowy- lista inicjalizacyjna

## Parametry

<code>_numer</code>	numer sali reprezentowany w klasie
<code>_rzad</code>	rzad -  -
<code>_kolumna</code>	kolumna -  -

## 3.7.2.3 Sala() [3/4]

```
Sala::Sala (
    int _numer )
```

Konstruktor jednoargumentowy- lista inicjalizacyjna

## Parametry

<code>_numer</code>	numer sali reprezentowany w klasie
---------------------	------------------------------------

## 3.7.2.4 Sala() [4/4]

```
Sala::Sala (
    const Sala & obj )
```

Przeładowany konstruktor kopiujący

## Parametry

<code>obj</code>	sala do skopiowania
------------------	---------------------

## 3.7.3 Dokumentacja funkcji składowych

## 3.7.3.1 operator=() [1/2]

```
Sala& Sala::operator= (
    const Sala && obj )
```

Przeładowany operator przeniesienia

**Parametry**

<i>obj</i>	sala do przeniesienia
------------	-----------------------

**Zwraca**

zwraca przeniesioną salę

**3.7.3.2 operator=()** [2/2]

```
Sala& Sala::operator= (
    const Sala & obj )
```

Przeładowany operator przypisania

**Parametry**

<i>obj</i>	sala do przypisania
------------	---------------------

**Zwraca**

zwraca przypisaną salę

**3.7.3.3 ustawKolumne()**

```
void Sala::ustawKolumne (
    int _kolumna )
```

Metoda typu setter, ustawiająca ilość kolumn

**Parametry**

<i>_kolumna</i>	ilość kolumn do ustawienia
-----------------	----------------------------

**3.7.3.4 ustawNumer()**

```
void Sala::ustawNumer (
    int _numer )
```

Metoda typu setter, ustawiająca numer sali



## Parametry

<code>_numer</code>	numer do ustawienia
---------------------	---------------------

**3.7.3.5 ustawRzad()**

```
void Sala::ustawRzad (
    int _rzad )
```

Metoda typu setter, ustawiająca ilość rzędów

## Parametry

<code>_rzad</code>	ilość rzędów do ustawienia
--------------------	----------------------------

**3.7.3.6 zwrocKolumne()**

```
int Sala::zwrocKolumne ( ) const
```

Metoda typu getter, zwracająca ilość kolumn

## Zwraca

ilość kolumn

**3.7.3.7 zwrocNumer()**

```
int Sala::zwrocNumer ( ) const
```

Metoda typu getter, zwracająca numer sali

## Zwraca

numer sali

### 3.7.3.8 zwrocPojemonsc()

```
int Sala::zwrocPojemonsc ( ) const
```

Metoda typu getter, zwracająca pojemność sali

**Zwraca**

ilość rzędów \* ilość kolumn

### 3.7.3.9 zwrocRzad()

```
int Sala::zwrocRzad ( ) const
```

Metoda typu getter, zwracająca ilość rzędów

**Zwraca**

ilość rzędów

## 3.7.4 Dokumentacja przyjaciół i funkcji związanych

### 3.7.4.1 operator<<

```
std::ostream& operator<< (
    std::ostream & os,
    const Sala & sala ) [friend]
```

Zaprzyjaźniony operator strumieniowy służący do wypisania sali do strumienia

**Parametry**

<i>os</i>	output stream
<i>sala</i>	sala do wypisania @retrun zwraca strumień

## 3.7.5 Dokumentacja atrybutów składowych

### 3.7.5.1 pNext

```
Sala* Sala::pNext
```

wskaźnik na następny element sali

Dokumentacja dla tej klasy została wygenerowana z pliku:

- kod/[Sala.hpp](#)



## Rozdział 4

# Dokumentacja plików

### 4.1 Dokumentacja pliku kod/Bilet.hpp

```
#include "Sala.hpp"  
#include "Data.hpp"  
#include "Status.h"  
#include <string>
```

Wykres zależności załączania dla Bilet.hpp:

### 4.2 Dokumentacja pliku kod/Data.hpp

```
#include <ostream>
```

Wykres zależności załączania dla Data.hpp: Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:

#### Komponenty

- class [Data](#)

### 4.3 Dokumentacja pliku kod/Plik.hpp

```
#include "Rezerwacje.hpp"  
#include <vector>
```

Wykres zależności załączania dla Plik.hpp:

#### Komponenty

- class [Plik](#)

#### Funkcje

- `std::vector< std::string > odczytajZKonsoli (const std::string &napis)`

### 4.3.1 Dokumentacja funkcji

#### 4.3.1.1 odczytajZKonsoli()

```
std::vector<std::string> odczytajZKonsoli (
    const std::string & napis )
```

Funkcja pomocnicza służąca do zczytania napisu z konsoli do wektora

##### Parametry

<i>napis</i>	napis do wczytania
--------------	--------------------

##### Zwraca

wektor napisów

## 4.4 Dokumentacja pliku kod/Rezerwacja.hpp

```
#include <iostream>
#include "Bilet.hpp"
#include <vector>
```

Wykres zależności załączania dla Rezerwacja.hpp: Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:

### Komponenty

- class [Rezerwacja](#)

## 4.5 Dokumentacja pliku kod/Rezerwacje.hpp

```
#include "Rezerwacja.hpp"
#include "Sala.hpp"
#include <fstream>
```

Wykres zależności załączania dla Rezerwacje.hpp: Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:

### Komponenty

- class [ListaRezerwacji](#)
- class [ListaSal](#)

## 4.6 Dokumentacja pliku kod/Sala.hpp

```
#include <ostream>
```

Wykres zależności załączania dla Sala.hpp: Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:

### Komponenty

- class `Sala`

## 4.7 Dokumentacja pliku kod/Status.h

Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:

### Wyliczenia

- enum `Status` { `Status::DOSTEPNY` = 0, `Status::REZERWACJA`, `Status::SPRZEDANY` }

### 4.7.1 Dokumentacja typów wyliczanych

#### 4.7.1.1 Status

```
enum Status [strong]
```

Typ wyliczeniowy służący do reprezentowania statusu biletu

Wartości wyliczeń

DOSTEPNY	bilet dostępny
REZERWACJA	zarezerwowany
SPRZEDANY	sprzedany





# Indeks

- ~Data
  - Data, [13](#)
- ~ListaRezerwacji
  - ListaRezerwacji, [19](#)
- ~ListaSal
  - ListaSal, [24](#)
- ~Rezerwacja
  - Rezerwacja, [29](#)
- Bilet, [5](#)
  - Bilet, [6](#)
  - operator<<, [11](#)
  - operator=, [7](#)
  - ustawDate, [7](#)
  - ustawID, [7](#)
  - ustawKolumne, [7](#)
  - ustawNazweFilmu, [8](#)
  - ustawRzad, [8](#)
  - ustawSale, [8](#)
  - ustawStatus, [9](#)
  - wypisz, [9](#)
  - zwrocDate, [9](#)
  - zwrocID, [9](#)
  - zwrocKolumne, [9](#)
  - zwrocNazweFilmu, [10](#)
  - zwrocRzad, [10](#)
  - zwrocSale, [10](#)
  - zwrocStatus, [10](#)
  - zwrocStatusJakoNapis, [11](#)
- czyDatyKoliduja
  - Data, [13](#)
- czyKoniecMiesiaca
  - Data, [13](#)
- Data, [11](#)
  - ~Data, [13](#)
  - czyDatyKoliduja, [13](#)
  - czyKoniecMiesiaca, [13](#)
  - Data, [12](#)
  - operator<<, [17](#)
  - operator>>, [17](#)
  - operator>=, [15](#), [17](#)
  - operator++, [14](#)
  - operator=, [14](#)
  - operator==, [14](#)
  - ustawDzien, [15](#)
  - ustawMiesiac, [15](#)
  - ustawRok, [16](#)
  - zwrocDzien, [16](#)
  - zwrocMiesiac, [16](#)
  - zwrocRok, [16](#)
  - dodajNaKoniec
    - ListaRezerwacji, [19](#)
  - dodajNaKoniecListy
    - ListaRezerwacji, [19](#)
    - ListaSal, [24](#)
  - dodajNaKoniecListyGut
    - ListaSal, [25](#)
  - DOSTEPNY
    - Status.h, [43](#)
  - ileWolnychKrsesel
    - ListaRezerwacji, [20](#)
  - instrukcja
    - Plik, [27](#)
  - kod/Bilet.hpp, [41](#)
  - kod/Data.hpp, [41](#)
  - kod/Plik.hpp, [41](#)
  - kod/Rezerwacja.hpp, [42](#)
  - kod/Rezerwacje.hpp, [42](#)
  - kod/Sala.hpp, [43](#)
  - kod/Status.h, [43](#)
  - kupBilet
    - ListaRezerwacji, [20](#)
  - ListaRezerwacji, [18](#)
    - ~ListaRezerwacji, [19](#)
    - dodajNaKoniec, [19](#)
    - dodajNaKoniecListy, [19](#)
    - ileWolnychKrsesel, [20](#)
    - kupBilet, [20](#)
    - ListaRezerwacji, [18](#), [19](#)
    - pHead, [23](#)
    - usunListe, [21](#)
    - usunRezerwacje, [21](#)
    - wypisz, [21](#)
    - wypiszWszystkieRezerwacje, [21](#)
    - wypiszZarezerwowaneBilety, [22](#)
    - zarezerwuj, [22](#)
    - zwrocIloscRezerwacji, [22](#)
    - zwrocRezerwacjeNumer, [23](#)
  - ListaSal, [23](#)
    - ~ListaSal, [24](#)
    - dodajNaKoniecListy, [24](#)
    - dodajNaKoniecListyGut, [25](#)
    - ListaSal, [24](#)
    - pHead, [26](#)
    - usunListe, [25](#)

- wypiszListe, 25
- zwrocIloscSal, 25
- zwrocSaleNumer, 25
- zwrocTakaSamaSale, 26
- odczytajZKonsoli
  - Plik.hpp, 42
- odczytajZPliku
  - Plik, 27
- operator<<
  - Bilet, 11
  - Data, 17
  - Rezerwacja, 33
  - Sala, 38
- operator>>
  - Data, 17
- operator>=
  - Data, 15, 17
- operator++
  - Data, 14
- operator=
  - Bilet, 7
  - Data, 14
  - Rezerwacja, 29
  - Sala, 35, 36
- operator==
  - Data, 14
- pHead
  - ListaRezerwacji, 23
  - ListaSal, 26
- Plik, 26
  - instrukcja, 27
  - odczytajZPliku, 27
  - zapiszDoPliku, 27
- Plik.hpp
  - odczytajZKonsoli, 42
- pNext
  - Rezerwacja, 33
  - Sala, 38
- REZERWACJA
  - Status.h, 43
- Rezerwacja, 28
  - ~Rezerwacja, 29
  - operator<<, 33
  - operator=, 29
  - pNext, 33
  - Rezerwacja, 29
  - ustawDate, 30
  - ustawLiczbeGosci, 30
  - ustawNazwe, 30
  - ustawSale, 30
  - ustawStatusNaFotelu, 31
  - wypisz, 31
  - zwrocBilet, 31
  - zwrocDate, 31
  - zwrocIleSprzedanych, 32
  - zwrocIleSprzedanychIZarezerwowanych, 32
  - zwrocIloscWizyt, 32
  - zwrocNazwe, 32
  - zwrocSale, 32
- Sala, 33
  - operator<<, 38
  - operator=, 35, 36
  - pNext, 38
  - Sala, 34, 35
  - ustawKolumne, 36
  - ustawNumer, 36
  - ustawRzad, 37
  - zwrocKolumne, 37
  - zwrocNumer, 37
  - zwrocPojemonsc, 37
  - zwrocRzad, 38
- SPRZEDANY
  - Status.h, 43
- Status
  - Status.h, 43
- Status.h
  - DOSTEPNY, 43
  - REZERWACJA, 43
  - SPRZEDANY, 43
  - Status, 43
- ustawDate
  - Bilet, 7
  - Rezerwacja, 30
- ustawDzien
  - Data, 15
- ustawID
  - Bilet, 7
- ustawKolumne
  - Bilet, 7
  - Sala, 36
- ustawLiczbeGosci
  - Rezerwacja, 30
- ustawMiesiac
  - Data, 15
- ustawNazwe
  - Rezerwacja, 30
- ustawNazweFilmu
  - Bilet, 8
- ustawNumer
  - Sala, 36
- ustawRok
  - Data, 16
- ustawRzad
  - Bilet, 8
  - Sala, 37
- ustawSale
  - Bilet, 8
  - Rezerwacja, 30
- ustawStatus
  - Bilet, 9
- ustawStatusNaFotelu
  - Rezerwacja, 31
- usunListe

- ListaRezerwacji, [21](#)
- ListaSal, [25](#)
- usunRezerwacje
  - ListaRezerwacji, [21](#)
- wypisz
  - Bilet, [9](#)
  - ListaRezerwacji, [21](#)
  - Rezerwacja, [31](#)
- wypiszListe
  - ListaSal, [25](#)
- wypiszWszystkieRezerwacje
  - ListaRezerwacji, [21](#)
- wypiszZarezerwowaneBilety
  - ListaRezerwacji, [22](#)
- zapiszDoPliku
  - Plik, [27](#)
- zarezerwuj
  - ListaRezerwacji, [22](#)
- zwrocBilet
  - Rezerwacja, [31](#)
- zwrocDate
  - Bilet, [9](#)
  - Rezerwacja, [31](#)
- zwrocDzien
  - Data, [16](#)
- zwrocID
  - Bilet, [9](#)
- zwrocIleSprzedanych
  - Rezerwacja, [32](#)
- zwrocIleSprzedanychIZarezerwowanych
  - Rezerwacja, [32](#)
- zwrocIloscRezerwacji
  - ListaRezerwacji, [22](#)
- zwrocIloscSal
  - ListaSal, [25](#)
- zwrocIloscWizyt
  - Rezerwacja, [32](#)
- zwrocKolumne
  - Bilet, [9](#)
  - Sala, [37](#)
- zwrocMiesiac
  - Data, [16](#)
- zwrocNazwe
  - Rezerwacja, [32](#)
- zwrocNazweFilmu
  - Bilet, [10](#)
- zwrocNumer
  - Sala, [37](#)
- zwrocPojemonsc
  - Sala, [37](#)
- zwrocRezerwacjeNumer
  - ListaRezerwacji, [23](#)
- zwrocRok
  - Data, [16](#)
- zwrocRzad
  - Bilet, [10](#)
  - Sala, [38](#)
- zwrocSale
  - Bilet, [10](#)
  - Rezerwacja, [32](#)
- zwrocSaleNumer
  - ListaSal, [25](#)
- zwrocStatus
  - Bilet, [10](#)
- zwrocStatusJakoNapis
  - Bilet, [11](#)
- zwrocTakaSamaSale
  - ListaSal, [26](#)