Lista dwukierunkowa

Generated by Doxygen 1.12.0

Chapter 1

Class Index

1.1 Class List

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

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

Chapter 3

Class Documentation

3.1 kom Class Reference

Klasa kom reprezentujaca komorke/wezel.

`#include <kom.h>`

Public Member Functions

- kom (int a)

  Konstruktor z parametrem.

Public Attributes

- int a

  Wartosc przechowywana w komorce listy.

- kom * next

  Wskaznik na nastepny element listy.
- kom * prev

  Wskaznik na nastepny poprzedni listy.

3.1.1 Detailed Description

Klasa kom reprezentujaca komorke/wezel.

Definition at line 10 of file kom.h.

3.1.2 Constructor & Destructor Documentation

3.1.2.1 kom()

```
kom::kom (
            int a)
```

Konstruktor z parametrem.

Parameters

a	Wartosc, ktora nalezy przypisac do pola a.

Definition at line 5 of file kom.cpp.

3.1.3 Member Data Documentation

3.1.3.1 a

```
int kom::a
```

Wartosc przechowywana w komorce listy.

Definition at line 12 of file kom.h.

3.1.3.2 next

```
kom* kom::next
```

Wskaznik na nastepny element listy.

Definition at line 13 of file kom.h.

3.1.3.3 prev

```
kom* kom::prev
```

Wskaznik na nastepny poprzedni listy.

Definition at line 14 of file kom.h.

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

- Projekt1zaawansowane/kom.h
- Projekt1zaawansowane/kom.cpp

3.2 listadk Class Reference

Klasa listadk reprezentujaca liste dwukierunkowa.

```
#include <kom.h>
```

Public Member Functions

- listadk (int a)

  Konstruktor listy dwukierunkowej, ktory tworzy liste z jednym elementem.
- ~listadk (void)

  Destruktor listy. Usuwa wszystkie elementy listy i zwalnia pamiec.
- void dodajglowa (int a)

  Dodaje nowy element o wartosci a na poczatku listy (przed glowa).
- void dodajogon (int a)

  Dodaje nowy element o wartosci a na końcu listy (za ogonem).
- void minusglowa (void)

  Usuwa element z poczatku listy (glowy).
- void minusogon (void)

  Usuwa element z końca listy (ogona).
- void dodajindex (int a, int b)

  Dodaje nowy element o wartosci a na pozycji b.
- void minusindex (int a)

  Usuwa element z podanego indeksu.
- void wypisz (void)

  Wypisuje zawartosc listy od glowy do ogona.
- void wypisztyl (void)

  Wypisuje zawartosc listy od ogona do glowy.
- void wypisznext (int a)

  Wypisuje wartosc elementu, ktory znajduje sie po elemencie o wartosci a.
- void wypiszprev (int a)

  Wypisuje wartosc elementu, ktory znajduje sie przed elementem o wartosci a.
- void usun (void)

  Usuwa wszystkie elementy z listy.

Private Attributes

- kom * glowa

  Wskaznik na pierwszy element listy (glowa).
- kom * ogon

  Wskaznik na ostatni element listy (ogon).

3.2.1 Detailed Description

Klasa listadk reprezentujaca liste dwukierunkowa.

Definition at line 24 of file kom.h.

3.2.2 Constructor & Destructor Documentation

3.2.2.1 listadk()

```
listadk::listadk (
            int a)
```

Konstruktor listy dwukierunkowej, ktory tworzy liste z jednym elementem.

Parameters

a	Wartosc pierwszego elementu listy.

Definition at line 6 of file kom.cpp.

3.2.2.2 ~listadk()

```
listadk::~listadk (
            void )
```

Destruktor listy. Usuwa wszystkie elementy listy i zwalnia pamiec.

Definition at line 7 of file kom.cpp.

3.2.3 Member Function Documentation

3.2.3.1 dodajglowa()

```
void listadk::dodajglowa (
            int a)
```

Dodaje nowy element o wartosci a na poczatku listy (przed glowa).

Parameters

a	Wartosc nowego elementu.

Definition at line 8 of file kom.cpp.

3.2.3.2 dodajindex()

```
void listadk::dodajindex (
            int a,
            int b)
```

Dodaje nowy element o wartosci a na pozycji b.

Parameters

a	Wartosc nowego elementu.
b	Indeks, na ktorym nalezy wstawic nowy element.

Definition at line 69 of file kom.cpp.

3.2.3.3 dodajogon()

```
void listadk::dodajogon (
            int a)
```

Dodaje nowy element o wartosci a na koñcu listy (za ogonem).

Parameters

a	Wartosc nowego elementu.

Definition at line 60 of file kom.cpp.

3.2.3.4 minusglowa()

```
void listadk::minusglowa (
            void )
```

Usuwa element z poczatku listy (glowy).

Definition at line 18 of file kom.cpp.

3.2.3.5 minusindex()

```
void listadk::minusindex (
            int a)
```

Usuwa element z podanego indeksu.

Parameters

a	Indeks elementu, ktory ma zostac usuniety.

Definition at line 28 of file kom.cpp.

3.2.3.6 minusogon()

```
void listadk::minusogon (
            void )
```

Usuwa element z koñca listy (ogona).

Definition at line 51 of file kom.cpp.

3.2.3.7 usun()

```
void listadk::usun (
            void )
```

Usuwa wszystkie elementy z listy.

Definition at line 129 of file kom.cpp.

3.2.3.8 wypisz()

```
void listadk::wypisz (
            void )
```

Wypisuje zawartosc listy od glowy do ogona.

Definition at line 90 of file kom.cpp.

3.2.3.9 wypisznext()

```
void listadk::wypisznext (
            int a)
```

Wypisuje wartosc elementu, ktory znajduje sie po elemencie o wartosci a.

Parameters

a	Wartosc elementu, po ktorym ma zostac wypisany nastepny element.

Definition at line 109 of file kom.cpp.

3.2.3.10 wypiszprev()

```
void listadk::wypiszprev (
            int a)
```

Wypisuje wartosc elementu, ktory znajduje sie przed elementem o wartosci a.

Parameters

a	Wartosc elementu, przed ktorym ma zostac wypisany poprzedni element.

Definition at line 119 of file kom.cpp.

3.2.3.11 wypisztyl()

```
void listadk::wypisztyl (
            void )
```

Wypisuje zawartosc listy od ogona do glowy.

Definition at line 103 of file kom.cpp.

3.2.4 Member Data Documentation

3.2.4.1 glowa

```
kom* listadk::glowa  [private]
```

Wskaznik na pierwszy element listy (glowa).

Definition at line 26 of file kom.h.

3.2.4.2 ogon

```
kom* listadk::ogon  [private]
```

Wskaznik na ostatni element listy (ogon).

Definition at line 27 of file kom.h.

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

- Projekt1zaawansowane/kom.h
- Projekt1zaawansowane/kom.cpp

Chapter 4

File Documentation

4.1 Projekt1zaawansowane/kom.cpp File Reference

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

4.2 kom.cpp

Go to the documentation of this file.

```
00001 #include "kom.h"
00002 #include <iostream>
00003 using namespace std;
00004
00005     kom::kom(int a) : a(a), next(nullptr), prev(nullptr) {}
00006     listadk::listadk(int a) { glowa = new kom(a),ogon=glowa; }
00007     listadk::~listadk(void) { this->usun(); }
00008     void listadk::dodajglowa(int a) {
00009         kom* komn = new kom(a);
00010         if (!glowa) { glowa = ogon = komn; }
00011         else {
00012             komn->next = glowa;
00013             glowa->prev = komn;
00014             glowa = komn;
00015
00016         }
00017     }
00018     void listadk::minusglowa(void) {
00019         if (!glowa) {
00020             cout « "brak glowy\n";
00021             return; }
00022         glowa = glowa->next;
00023         if (glowa) {
00024             glowa->prev = nullptr;
00025         }
00026         else { ogon = nullptr; }
00027     }
00028     void listadk::minusindex(int a){
00029         if (!glowa) {
00030             cout « "brak glowy\n";
00031             return;
00032         }
00033         if (a == 0) {
00034             this->minusglowa();
00035             return;
00036         }
00037         kom* o = glowa;
00038         for (int i = 0;o != nullptr && i < a;i++) { o = o->next; }
00039         if (o == nullptr) {
00040             this->minusogon();
00041             return;
00042         }
```

```
00043         if (o->prev != nullptr) {
00044             o->prev->next = o->next;
00045         }
00046         if (o->next != nullptr) {
00047             o->next->prev = o->prev;
00048         }
00049         delete o;
00050         };
00051     void listadk::minusogon(void) {
00052         if (!ogon) { cout « "brak ogon\n"; }
00053         ogon = ogon->prev;
00054         if (ogon) {
00055             ogon->next = nullptr;
00056         }
00057         else { glowa = nullptr; }
00058 
00059     }
00060     void listadk::dodajogon(int a) {
00061         kom* komn = new kom(a);
00062         if (!ogon) { glowa = ogon = komn; }
00063         else {
00064             komn->prev = ogon;
00065             ogon->next = komn;
00066             ogon = komn;
00067         }
00068     }
00069     void listadk::dodajindex(int a, int b) {
00070         kom* komn = new kom(a);
00071         if (b == 0) {
00072             dodajglowa(a);
00073             return;
00074         }
00075         kom* o = glowa;
00076         for (int i = 0;o != nullptr && i < b;i++) { o = o->next; }
00077         if (o == nullptr) { dodajogon(a); }
00078         else {
00079             komn->next = o->next;
00080             komn->prev = o;
00081             o->next = komn;
00082             if (komn->next != nullptr) {
00083                 komn->next->prev = komn;
00084             }
00085             else {
00086                 ogon = komn;
00087             }
00088         }
00089     }
00090     void listadk::wypisz(void) {
00091         if (!glowa) {
00092             cout « "brak glowy\n";
00093             return;
00094         }
00095         kom* o = glowa;
00096         cout « endl;
00097         for (int i = 0;o != nullptr;i++) {
00098         cout« o->a « "  ";
00099         o = o->next;
00100         }
00101         cout « endl;
00102     }
00103     void listadk::wypisztyl(void) {
00104         kom* o = ogon;
00105         cout « endl;
00106         for (int i = 0;o != nullptr;i++) { cout « o->a « "  ", o = o->prev; }
00107         cout « endl;
00108     }
00109     void listadk::wypisznext(int a) {
00110         if (!glowa) {
00111             cout « "brak glowy\n";
00112             return;
00113         }
00114         kom* o = glowa;
00115         for (int i = 0;o != nullptr && i < a;i++) { o = o->next; }
00116         kom* s = o->next;
00117         cout « s->a«endl;
00118     }
00119     void listadk::wypiszprev(int a){
00120         if (!glowa) {
00121             cout « "brak glowy\n";
00122             return;
00123         }
00124         kom* o = glowa;
00125         for (int i = 0;o != nullptr && i < a;i++) { o = o->next; }
00126         kom* s = o->prev;
00127         cout « s->a«endl;
00128     }
00129     void listadk::usun(void) {
```

```
00130         while (glowa != nullptr) {
00131            minusglowa();
00132         }
00133     }
```

4.3 Projekt1zaawansowane/kom.h File Reference

Classes

- class kom

  Klasa kom reprezentujaca komorke/wezel.
- class listadk

  Klasa listadk reprezentujaca liste dwukierunkowa.

4.4 kom.h

Go to the documentation of this file.

```
00001 #pragma once
00010 class kom {
00011 public:
00012     int a;
00013     kom* next;
00014     kom* prev;
00019     kom(int a);
00020 };
00024 class listadk {
00025 private:
00026     kom* glowa;
00027     kom* ogon;
00028
00029 public:
00034     listadk(int a);
00038     ~listadk(void);
00043     void dodajglowa(int a);
00048     void dodajogon(int a);
00052     void minusglowa(void);
00056     void minusogon(void);
00062     void dodajindex(int a, int b);
00067     void minusindex(int a);
00071     void wypisz(void);
00075     void wypisztyl(void);
00080     void wypisznext(int a);
00085     void wypiszprev(int a);
00089     void usun(void);
00090 };
```

4.5 Projekt1zaawansowane/main.cpp File Reference

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

Functions

- int main ()

4.5.1 Function Documentation

4.5.1.1 main()

```
int main ()
```

Definition at line 5 of file main.cpp.

4.6 main.cpp

Go to the documentation of this file.

```
00001 #include <iostream>
00002 #include "kom.h"
00003 using namespace std;
00004
00005 int main()
00006 {
00007     cout « "tworzenie listy z 5: ";
00008     listadk a(5);
00009     a.wypisz();
00010     cout « "dodanie 8 do przodu listy: ";
00011     a.dodajglowa(8);
00012     a.wypisz();
00013     cout « "dodanie 3 do tylu listy: ";
00014     a.dodajogon(3);
00015     a.wypisz();
00016     cout « "dodanie 4x4 do tylu listy: ";
00017     a.dodajogon(4);
00018     a.dodajogon(4);
00019     a.dodajogon(4);
00020     a.dodajogon(4);
00021     a.wypisz();
00022     cout « "dodanie 1 na indeksie 3: ";
00023     a.dodajindex(1, 2);
00024     a.wypisz();
00025     cout « "usunięcie z przodu listy: ";
00026     a.minusglowa();
00027     a.wypisz();
00028     cout « "usunięcie z tylu listy: ";
00029     a.minusogon();
00030     a.wypisz();
00031     cout « "wyswietlenie listy od przody i od tylu: ";
00032     a.wypisz();
00033     a.wypisztyl();
00034     cout « "wyswietlenie nastepnego elementu o numerze  indexu 5: ";
00035     a.wypisz();
00036     a.wypisznext(4);
00037     cout « "wyswietlenie poprzedniego elementu o numerze  indexu 2: ";
00038     a.wypisz();
00039     a.wypiszprev(1);
00040     cout « "Czyszczenie listy: ";
00041     a.usun();
00042     a.wypisz();
00043 }
```

Index