|  |
| --- |
| **2019년 10월 31일 실습보고서** |
| **7조 조윤직 송재원 양석준 진영** |
| **실습자료1 : [stack,que]** |
| **소스코드** |
| //cque.h  #pragma once  #include "clinkedlist.h"  class cque  {  private:  clinkedlist list;  public:  cque();  ~cque();  void push(const int& key);  int pop(bool& suc);  };  //cque.cpp  #include "cque.h"  cque::cque()  {  }  cque::~cque()  {  }  void cque::push(const int& key)  {  list.insert3(key);  }  int cque::pop(bool & suc)  {  cnode\* fp = list.gethead();  cnode\* p = fp->getNext();  if (p != list.gettail()) {  fp->setNext(p->getNext());  int num = p->getkey();  delete p;  suc = true;  return num;  }  else {  suc = false;  return 0;  }  }  //clinkedlist.h  #pragma once  #include "cnode.h"  class clinkedlist  {  private:  cnode\* head;  cnode\* tail;  public:  clinkedlist();  ~clinkedlist();  void insert1(const int& key); // 헤드뒤에 추가  void showlist();  void deleteall();  void insert2(const int& key); // 정렬되서 추가  cnode\* findnode(const int& key);  int getkey(cnode& c);  void delete\_key(const int& key);  void insert3(const int& key); // 테일 앞에 추가  cnode\* gethead();  cnode\* gettail();  };  //clinkedlist.cpp  #include "clinkedlist.h"  #include <iostream>  using namespace std;  clinkedlist::clinkedlist()  {  head = new cnode();  tail = new cnode();  head->setNext(tail);  head->setbefore(head);  tail->setNext(tail);  tail->setbefore(head);  }  clinkedlist::~clinkedlist()  {  deleteall();  delete head;  delete tail;  }  void clinkedlist::insert1(const int& key)  {  cnode\* temp = new cnode();  temp->setkey(key);  temp->setNext(head->getNext());  head->setNext(temp);  }  void clinkedlist::showlist()  {  cnode\* p = head->getNext();  while (p != tail) {  cout << p->getkey() << endl;  p = p->getNext();  }  }  void clinkedlist::deleteall()  {  cnode\* p = head->getNext();  while (p != tail) {  cnode\* s = p;  p = p->getNext();  delete s;  }  head->setNext(tail);  }  void clinkedlist::insert2(const int& key)  {  cnode\* fp = head;  cnode\* p = head->getNext();  cnode\* temp = new cnode;  temp->setkey(key);  while (p != tail) {  if (key < p->getkey()) {  break;  }  else {  fp = p;  p = p->getNext();  }  }  fp->setNext(temp);  temp->setNext(p);  }  cnode \* clinkedlist::findnode(const int& key)  {  cnode\* s = head->getNext();  while (s != tail && s->getkey() != key) {  s = s->getNext();  }  return s;  }  int clinkedlist::getkey(cnode &c)  {  return c.getkey();  }  void clinkedlist::delete\_key(const int& key)  {  cnode\* fp = head;  cnode\* s = fp->getNext();  while (s != tail && s->getkey() != key) {  fp = s;  s = fp->getNext();  }  if (s != tail) {  fp->setNext(s->getNext());  cout << s->getkey() << "삭제함" << endl;  delete s;  }  }  void clinkedlist::insert3(const int& key)  {  // cnode\* temp = new cnode();  // temp->setkey(key);  // temp->setNext(head->getNext());  // head->setNext(temp);  cnode\* fp = head;  cnode\* s = fp->getNext();  cnode\* p = new cnode;  p->setkey(key);  while (s != tail) {  fp = s;  s = fp->getNext();  }  fp->setNext(p);  p->setNext(tail);  }  cnode \* clinkedlist::gethead()  {  return head;  }  cnode \* clinkedlist::gettail()  {  return tail;  }  //cstack.h  #pragma once  #include "clinkedlist.h"  class cstack  {  private:  clinkedlist list;  public:  cstack();  ~cstack();  void push(const int& key);  int pop(bool& suc);  };  //cstack.cpp  #include "cstack.h"  #include <iostream>  using namespace std;  cstack::cstack()  {  }  cstack::~cstack()  {  }  void cstack::push(const int& key)  {  list.insert1(key);  }  int cstack::pop(bool& suc)  {  cnode\* p = list.gethead();  cnode\* pp = p->getNext();  if (pp != list.gettail()) {  p->setNext(pp->getNext());  int num = pp->getkey();  delete pp;  suc = true;  return num;  }  else {  suc = false;  return 0;  }  }  //cnode.h  #pragma once  class cnode  {  private:  int key;  cnode\* next;  cnode\* before;  public:  cnode();  ~cnode();  int getkey() {  return key;  }  void setkey(const int& key) {  this->key = key;  }  cnode\* getNext() {  return next;  }  cnode\* getbefore() {  return before;  }  void setbefore(cnode\* before) {  this->before = before;  }  void setNext(cnode\* next) {  this->next = next;  }  };  //cnode.cpp  #include "cnode.h"  cnode::cnode()  {  }  cnode::~cnode()  {  }  //main.cpp  #include "clinkedlist.h"  #include "cstack.h"  #include "cque.h"  #include <iostream>  using namespace std;  int main() {  /\*clinkedlist list;  list.insert1(10);  list.insert1(20);  list.insert1(30);  list.insert1(40);  list.showlist();  cout << endl << endl;  list.insert3(77);  list.insert3(88);  list.showlist();\*/  cout << "c++ 7조 송재원, 조윤직, 양석준, 박진영입니다." << endl << "stack" << endl;  cstack j;  bool TF = false;  int n;  n = j.pop(TF);  if (TF)  cout << n << endl;  j.push(1);  j.push(2);  j.push(3);  j.push(4);  n = j.pop(TF);  if (TF)  cout << n << endl;  j.push(7);  n = j.pop(TF);  if (TF)  cout << n << endl;  n = j.pop(TF);  if (TF)  cout << n << endl;  n = j.pop(TF);  if (TF)  cout << n << endl;  n = j.pop(TF);  if (TF)  cout << n << endl;  n = j.pop(TF);  if (TF)  cout << n << endl;  n = j.pop(TF);  if (TF)  cout << n << endl;  n = j.pop(TF);  if (TF)  cout << n << endl;  cout << "Que" << endl;  cque j1;  TF = false;  n = 0;  n = j1.pop(TF);  if (TF)  cout << n << endl;  j1.push(1);  j1.push(2);  j1.push(3);  j1.push(4);  n = j1.pop(TF);  if (TF)  cout << n << endl;  j1.push(7);  n = j1.pop(TF);  if (TF)  cout << n << endl;  n = j1.pop(TF);  if (TF)  cout << n << endl;  n = j1.pop(TF);  if (TF)  cout << n << endl;  n = j1.pop(TF);  if (TF)  cout << n << endl;  n = j1.pop(TF);  if (TF)  cout << n << endl;  n = j1.pop(TF);  if (TF)  cout << n << endl;  n = j1.pop(TF);  if (TF)  cout << n << endl;  } |
| **실행결과** |
|  |