חורף 2011 מועד א - פתרון

1. א.

#include <string.h>

#include <stdlib.h>

#include <stdio.h>

#include "set.h"

typedef struct DayLate {

int day;

int minutes;

char\* excuse;

} DayLate;

DayLate\* createLate(int day, int minutes, char\* excuse){

DayLate\* res = malloc(sizeof(DayLate));

int excuseLen = strlen(excuse);

if (! res) return NULL;

res->excuse= malloc(excuseLen +1);

if (! excuse) {

free(res);

return NULL;

}

strncpy(res->excuse, excuse, excuseLen+1);

res->day = day;

res->minutes = minutes;

return res;

}

///////////////////code for the students to write - 12 points////////////////////

//key function - total 2 points

int getDay(SetElementP elem) {

DayLate\* daylate = (DayLate\*) elem;

if (! daylate)

return 0;

return daylate->day; //return day - 2 points

}

//delete function - total 4 points

void deleteDay(SetElementP elem) {

DayLate\* daylate = (DayLate\*) elem;

if (! daylate)

return;

free(daylate->excuse); //free excuse - 2 points

free(daylate); // free daylate - 2 points

}

//hadndeling failed allocations - either in deleteDay / getDay or main - 2 points

//main - total 4 points

int main(){

SetStructureP dannyLateSet = createSet(365, getDay, deleteDay); //createset 1 point

SetStructureP yossyLateSet = createSet(365, getDay, deleteDay);

addElement(dannyLate, createLate(35,5,"Bus was late")); //addElement 1 point

addElement(dannyLate, createLate(67,5,"Bus was late"));

addElement(yossyLate, createLate(186,150,"I thought it was Saturday"));

printf("danny late days: %d\n", getElementCount(dannyLate)); //getElementCount 1 point

printf("yossy late days: %d\n", getElementCount(yossyLate));

destroySet(dannyLate); //destroySet 1 point

destroySet(yossyLate);

return 0;

}

ב.

SetElementP getElement(SetStructureP set, int key){

int i;

if (!set) //check input - 1 point

return NULL;

for(i=0;i<set->numOfElements;i++) { //loop - 2 points

if (set->getKey(set->elements[i]) == key) //use getKey - 1 point

return set->elements[i]; //return value - 1point

}

return NULL; //NULL return values - 1 point

}

//addElement - 6 points

result addElement(SetStructureP set, SetElementP element){

int elKey;

//null checks - 1 point

//MAX\_ELEMENTS check - 1 point

if (!set || !element || set->numOfElements >= MAX\_ELEMENTS)

return FAILURE;

elKey = set->getKey(element); //check existing element - 3 points

if (getElement(set,elKey))

return FAILURE;

set->elements[set->numOfElements++] = element; //increaseNumOfElements - 1 point

return SUCCESS;

}

//getElementCount - 3 points

int getElementCount(SetStructureP set){

return set->numOfElements;

}

//destroySet - 3 easy points

void destroySet(SetStructureP set){

destroySet\_old(set);

}

2.

א. C++ רגיל

**1.א.**

template <class Key, class Val, unsigned int \_nItems>

class Dictionary {

/\*...\*/

public:

Dictionary();

virtual ~Dictionary();

Val\* operator[](const Key& key) const;

bool RemoveVal(const Key& key, const Val& val);

bool AddVal(const Key& key, const Val& val);

};

template <class Key, class Val, unsigned int \_nItems>

ostream& operator<<(ostream& os, const Dictionary<Key, Val, \_nItems>&);

**ב.**

הטיפוס חייב לתמוך בהשמה (מעבירים רפרנסים שאמורים להיות מועתקים ע"י המבנה), בקונסטראקטור דיפולטי.(מחזירים מערך של values) ובאופרטור הדפסה.

**2.**

**א.**

class Dorms

{

protected:

static unsigned long \_nTotalResidents;

char\* \_dormsName;

char\* \_dormsAdmin;

public:

Dorms(const char\* dormsName, const char\* administrator);

virtual ~Dorms();

virtual bool FlatHasVacancy(const DormAddress& addr) const = 0;

static unsigned long GetTotalResidents() { return \_nTotalResidents; }

};

class FamilyDorms: public Dorms

{

private:

Dictionary<DormAddress, Resident, 1> \_data;

public:

FamilyDorms(const char\* dormsName, const char\* administrator) : Dorms(dormsName, administrator) { };

void FreeFlat(const DormAddress& addr);

bool FlatHasVacancy(const DormAddress& addr) const;

void PopulateFlat(const DormAddress& addr, const Resident& resident);

};

template <unsigned int \_nResidents>

class SinglesDorms: public Dorms

{

private:

Dictionary<DormAddress, Resident, \_nResidents> \_data;

public:

SinglesDorms(const char\* dormsName, const char\* administrator) : Dorms(dormsName, administrator) { };

void RemoveResident(const DormAddress& addr);

void AddResident(const DormAddress& addr, const Resident& resident);

bool FlatHasVacancy(const DormAddress& addr) const;

};

**ג.**

יש להגדיר מתודה ורטואלית טהורה נוספת:

virtual void Dorms::PrintResidents(ostream& os) const = 0;

ושני ממושיה במחלקות הנגזרות:

void SinglesDorms::PrintResidents(ostream& os) const;

void FamilyDorms::PrintResidents(ostream& os) const;

ולהגדיר אופרטור גלובאלי:

ostream& operator<<(ostream& os, const Dorms& drm)

{

drm.PrintResidents(os);

return os;

}

C++ חלק ב: מה יודפס?

stage 1

G::G() H() G::G() K::K(char\*)

stage 2:

G::G() H() G::G() K::K(char\*)

stage 3:

G::G() H() G::G() K(K&)

stage 4:

G::G() K::f1() G::G(G&) G::op= G::~G()

stage 5:

G::G(G&) H::f2(G) G::~G()

stage 6:

H::f3()

stage 7:

G::~G() K::~K() G::~G() H::~H() G::~G() K::~K() G::~G() H::~H() G::~G()

שאלה 3: UNIX והנדסת תוכנה

3.

Bash

א.

double\_lines

cat source-file | double

double

#!/bin/bash

while read line; do

echo $line

echo $line

done

ב.

#!/bin/bash

for file in $1/\*; do

if [[ -f $file ]]; then

a=`cat $file|head -1|cut -c1-3`

if [[ $a == "abc" ]]; then

/bin/rm $file

fi

UML+DP

1. (השורות באדום )

class Singleton {

  static Singleton s;

  int i;

  Singleton(int x) : i(x) { }

  Singleton& operator=(Singleton&);  // Disallowed

  Singleton(const Singleton&);       // Disallowed

public:

  static Singleton& instance() { return s; }

  int getValue() { return i; }

  void setValue(int x) { i = x; }

};

ב (אין צורך לפרט את תוכן המחלקות, orderDetail ו- Item הוצאו מהשאלה)

