פתרון מועד א סמסטר חורף 2015

**שאלה 1 – ADT**

סעיף א'

// User function types

typedef BOOL (\*LESS\_THAN\_PR) (pPriority, pPriority);

typedef void (\*DEL\_PR) (pPriority);

typedef void (\*DEL\_DATA) (pData);

typedef struct PQ\_ {

pNode pHead;

LESS\_THAN\_PR lt\_pr\_f;

DEL\_PR del\_pr\_f;

DEL\_DATA del\_d\_f;

} PQ, \*pPQ;

סעיף ב'

pPQ CreatePQ(LESS\_THAN\_PR lt\_pr\_f, DEL\_PR del\_pr\_f, DEL\_DATA del\_d\_f);

סעיף ג'

pData PopElement(pPQ ps)

{

pNode ppop;

pData pd;

if (NULL == ps)

return NULL;

if (NULL != ps->pHead)

{

ppop = ps->pHead;

pd = ps->pHead->pd;

ps->pHead = ps->pHead->pnext;

ps->del\_pr\_f(ppop->pp);

free(ppop);

return pd;

}

else

return NULL;

}

סעיף ד'

Result InsertElement(pPQ ps, pPriority pp, pData pd)

{

pNode pcurr, pnew;

if ( (NULL == ps) || (NULL == pp) || (NULL == pd) )

return FAILURE;

// Creating the new element

pnew = (pNode)malloc( sizeof(Node) );

if (NULL == pnew)

return FAILURE;

pnew->pp = pp;

pnew->pd = pd;

if ( (NULL == ps->pHead) || ps->lt\_pr\_f(ps->pHead->pp, pp) )

{

// Inserting the new elemet to the first place

pnew->pnext = ps->pHead;

ps->pHead = pnew;

}

else

{

// Finding location of the new element

pcurr = ps->pHead;

while (NULL != pcurr->pnext)

{

if ( ps->lt\_pr\_f(pcurr->pnext->pp, pp) )

break;

else

pcurr = pcurr->pnext;

}

// Inserting the new element

pnew->pnext = pcurr->pnext;

pcurr->pnext = pnew;

}

return SUCCESS;

}

סעיף ה'

BOOL LessThan\_Priority(pPriority ppl, pPriority ppr)

{

if ( \*( (int\*)ppl ) < \*( (int\*)ppr ) )

return TRUE;

else

return FALSE;

}

void DeletePriority(pPriority pp)

{

free(pp);

}

void DeleteTicket(pData pd)

{

free( ( (pTicket)pd )->message );

free(pd);

}

pq = CreatePQ(LessThan\_Priority, DeletePriority, DeleteTicket);

שאלה 2 C++: (30 נק')

א.

class File {

public:

File( std::string name = "" );

virtual ~File() {}

virtual std::string getName() const;

virtual void setName( std::string name );

virtual int getID() const;

virtual void print() const = 0;

protected:

int m\_ID;

std::string m\_name;

private:

static int s\_filesCount;

};

class TextFile : public File {

public:

TextFile( std::string name, std::string text = "" );

std::string getText() const;

void setText( std::string text );

void append( std::string text );

void print() const;

private:

std::string m\_text;

};

typedef std::vector<File\*> Files;

typedef std::vector<File\*>::iterator FilesIterator;

class Directory : public File {

public:

Directory( std::string name );

~Directory();

bool AddFile( File\* file );

bool RemoveFile( std::string name );

FilesIterator begin();

FilesIterator end();

int getNumFiles() const;

void print() const;

private:

Files m\_files;

};

ב. ההנחה כאן היא שהמשתמש אחראי על הקצאת הזיכרון, המחלקה מבצעת העתקה של המצביע בלבד ואחראית על שחרור הזכרון בסוף השימוש.

bool Directory::AddFile( File\* file )

{

for( unsigned i = 0; i < m\_files.size(); ++i )

{

if( m\_files[i]->getID() == file->getID() )

{

return false;

}

}

m\_files.push\_back( file );

return true;

}

bool Directory::RemoveFile( std::string name )

{

for( unsigned i = 0; i < m\_files.size(); ++i )

{

if( m\_files[i]->getName() == name )

{

delete m\_files[i];

m\_files.erase( m\_files.begin() + i);

return true;

}

}

return false;

}

ג.

File\* Find( Directory\* dir, string name )

{

FilesIterator itB = dir->begin();

FilesIterator itE = dir->end();

for( FilesIterator it = itB; it != itE; it++ )

{

if( (\*it)->getName() == name )

{

return \*it;

}

Directory\* t = dynamic\_cast<Directory\*>(\*it);

if( NULL != t )

{

File\* ret = Find( t, name );

if( NULL != ret )

{

return ret;

}

}

}

return NULL;

}

**שאלה 3 - מה יודפס? – פתרון**

stage 1:

A::A()

stage 2:

A::A(A&)

stage 3:

B::B(T)

stage 4:

A::A(A&)

A::A(A&)

B::B(T)

~A::A()

stage 5:

stage 6:

B::f()

A::A(A&)

~A::A()

stage 7:

A::A()

B::B()

C::C(A\*)

stage 8:

B::f()

A::A(A&)

~A::A()

stage 9:

B::g()

~C::C()

~B::B()

~A::A()

Caught Exception!

stage 10:

// nothing to print here

שאלה 4

חלק א: bash

calc1:

#!/bin/bash

cat exam-grades hmwk-grade | sort -n | ./calc2

calc2

#!/bin/bash

while read line; do

ls1=( $line )

read line

ls2=( $line )

(( sum=${ls1[1]}\*3/4+${ls2[1]}\*1/4+0 ))

echo ${ls1[0]} $sum

done

חלק ב': שאלות הבנה

נחוץ לטיפול באובייקטים const. אופרטור אחד פועל בווקטור שאינו קבוע, והשני מטפל בווקטורים מסוג const.

1. המצביע עטוף באובייקט אשר תמיד משחרר את הזיכרון שלו כאשר יוצאים מהבלוק, דרך הדיסטרקטור.
2. העברת פרמטר לcatch - by-ref תתפוס גם את החריגות שנורשו מexception - ולא "תחתוך" אותן.

אילו היינו מעבירים את exception על ידי ערך (כלומר by-val), החריגה הייתה "נחתכת" והייתה מדפיסה את המחרוזת “exception happened” בלי לתת אינדיקציה על החריגה האמתית שקרתה.