**אביב 2014 מועד א - פתרון**

1. **שאלת ADT – פתרון**

**סעיף א'**

typedef pKey (\*GetKey) (pElement);

typedef Bool (\*CompareKeys) (pKey, pKey);

typedef CompRes (\*CompareElements) (pElement, pElement);

typedef void (\*PrintElement) (pElement);

typedef void (\*DeleteElement) (pElement);

typedef struct \_SortedList {

pNode head;

unsigned int numElements;

GetKey GetKeyFunc;

CompareKeys CompareKeysFunc;

CompareElements CompareElementsFunc;

PrintElement PrintElementFunc;

DeleteElement DeleteElementFunc;

} SortedList;

**סעיף ב'**

pSList CreateSList (GetKey, CompareKeys, CompareElements, PrintElement, DeleteElement);

**סעיף ג'**

Result RemoveElement(pSList list, pKey key) {

pNode iter, temp;

if (!list || !key | !list->head)

return FAILURE;

iter = list->head;

if (TRUE == list->CompareKeysFunc(key, list->GetKeyFunc(iter->element))) {

list->head = iter->next;

free(iter);

return SUCCESS;

}

for (; iter->next != NULL; iter = iter->next) {

if (TRUE == list->CompareKeysFunc(key, list->GetKeyFunc(iter->next->element))) {

temp = iter->next;

iter->next = iter->next->next;

free(temp);

return SUCCESS;

}

}

return FAILURE;

};

**סעיף ד'**

pKey GetMovieName (pElement elem) {

pMovie movie = (pMovie) elem;

if (!movie)

return NULL;

return movie->name;

};

Bool CompareMovieNames (pKey key\_a, pKey key\_b) {

if (!key\_a || !key\_b)

return FALSE;

if (0 == strcmp((char\*) key\_a, (char\*) key\_b))

return TRUE;

return FALSE;

};

CompRes CompareMovies (pElement elem\_a, pElement elem\_b) {

pMovie movie\_a = (pMovie) elem\_a;

pMovie movie\_b = (pMovie) elem\_b;

if (!elem\_a)

return B\_BETTER;

if (!elem\_b)

return A\_BETTER;

if (movie\_a->score >= movie\_b->score)

return A\_BETTER;

return B\_BETTER;

};

void PrintMovie (pElement elem) {

pMovie movie = (pMovie) elem;

if (!elem)

return;

printf("( %3d ) %s (%d) - %s\n", movie->score, movie->name, movie->year, movie->director);

};

void DeleteMovie (pElement elem) {

pMovie movie = (pMovie) elem;

if (!elem)

return;

free(movie->name);

free(movie->director);

free(movie);

return;

};

1. **פתרון שאלת C++**

**סעיף א'**

template <class T, unsigned int size>

class Vector {

public:

Vector() {};

Vector(const Vector<T, size>& rhs);

Vector<T, size>& operator=(const Vector<T, size>& rhs);

T& operator()(unsigned int ind);

const T& operator()(unsigned int ind) const;

Vector<T, size> operator+(const Vector<T, size>& rhs);

T operator\*(const Vector<T, size>& rhs);

Vector<T, size> operator\*(const T& rhs);

Vector<T, size> operator-();

protected:

T arr\_[size];

};

**סעיף ב'**

template <class T, unsigned int size>

T& Vector<T, size>::operator()(unsigned int ind) {

return arr\_[ind];

}

template <class T, unsigned int size>

T Vector<T, size>::operator\*(const Vector<T, size>& rhs) {

T res = arr\_[0] \* rhs.arr\_[0];

for (int i = 1; i<size; i++)

res = res + arr\_[i] \* rhs.arr\_[i];

return res;

}

template <class T, unsigned int size>

Vector<T, size> Vector<T, size>::operator\*(const T& rhs) {

Vector<T, size> res;

for (int i = 0; i<size; i++)

res.arr\_[i] = arr\_[i]\*rhs;

return res;

}

template <class T, unsigned int size>

Vector<T, size> Vector<T, size>::operator-() {

Vector<T, size> res;

for (int i = 0; i<size; i++)

res.arr\_[i] = -arr\_[i];

return res;

}

**סעיף ג'**

class Complex {

public:

Complex(double a=0, double b=0): a\_(a), b\_(b) {};

Complex(const Complex& rhs): a\_(rhs.a\_), b\_(rhs.b\_) {};

Complex operator+(const Complex& rhs) {

Complex res;

res.a\_ = a\_ + rhs.a\_;

res.b\_ = b\_ + rhs.b\_;

return res;

}

Complex operator\*(const Complex& rhs) {

Complex res;

res.a\_ = a\_\*rhs.a\_ - b\_\*rhs.b\_;

res.b\_ = a\_\*rhs.b\_ + b\_\*rhs.a\_;

return res;

}

Complex operator-() {

Complex res;

res.a\_ = -a\_;

res.b\_ = -b\_;

return res;

}

protected:

double a\_;

double b\_;

};

1. **פתרון שאלת מה יודפס:**

**Stage 1:**

A::A()

B::B()

A::A()

B::B()

C::C()

**Stage 2:**

A::A()

B::B(int)

A::A()

B::B()

C::C()

D::D()

A::A(A&)

A::A()

B::B()

C::C()

D::D(D&)

**Stage 3:**

**Stage 4:**

D::f(int)

B::f(int)

B::f(int)

**Stage 5:**

A::g(int)

D::g(int)

B::g(int)

A::A()

B::B()

Caught A.

B::~B()

A::~A()

**Stage 6:**

B::h(B)

A::A(A&)

C::h(B)

A::A(A&)

**Stage 7:**

A::~A()

**Stage 8:**

A::~A()

A::~A()

D::~D()

C::~C()

B::~B()

A::~A()

B::~B()

A::~A()

C::~C()

B::~B()

A::~A()

B::~B()

A::~A()

1. **פתרון**
2. **פתרון שאלות הבנה**
3. Shared\_ptr משתף בעלות על המשאב

Unique\_ptr לוקח בעלות על המשאב.

1. נגדיר function object אשר מממש פעולת ">" אשר משווה בין שדות המחרוזת של האובייקטים.
2. האיטרטור שדרכו ניגשים לאיברי המיכל מבלי תלות במימוש המיכל, ואחדות הממשק של המיכלים השונים.
3. מוסיפים operator[] נוסף שמוגדר כמתודה const ומחזיר const reference לאיבר במערך.
4. **פתרון BASH**
5. Replace
6. #!/bin/bash
7. cat source | replace1 $1
8. replace1
9. #!/bin/bash
10. while read line; do
11. label=`echo $line | cut -f2 -d:`
12. if [[ $label =~ "CHANGE" ]]; then
13. a=`echo $line | cut -f1 -d:`
14. b=`echo $line | cut -f3 -d:`
15. echo $a $1 $b
16. else
17. echo $line
18. fi
19. done