חורף 2010/11 מועד ב - פתרון

**פתרון שאלה 1**

**א.**

typedef void Item;

typedef enum \_BOOL { FALSE, TRUE } BOOL;

typedef struct \_List List;

/\*Interface functions\*/

List\* CreateList();

void DesstroyList(List\*);

Item\* GetHead(List\*);

Item\* GetNext(List\*);

BOOL AddTail(List\*, Item\*);

Item\* GetTail(List\*);

**ב.**

typedef void StateData;

typedef struct \_State {

int \_stateId;

StateData\* \_stateData;

} State;

State\* CreateState(StateData\* stateData, int id);

void DestroyState(State\*);

int GetStateId(State\*);

/\*User Functions\*/

typedef State\* (\*ExecuteFunc)(State\* pInputState);

typedef struct \_Activity

{

State\* \_inputState;

State\* \_outputState;

ExecuteFunc \_executeFunc;

} Activity;

/\*Interface functions\*/

Activity\* CreateActivity(State\* in, State\* out, ExecuteFunc pExecuteFunc);

void DestroyActivity(Activity\*);

State\* GetInputState(Activity\* pAct);

void SetInputState(Activity\* pAct, State\* pState);

State\* GetOuputState(Activity\* pAct);

BOOL Execute(Activity\* pAct);

**ג.**

typedef struct \_Process

{

List\* \_activities;

State\* \_initState;

State\* \_finalState;

} Process;

Process\* CreateProcess();

void DestroyProcess(Process\* pPrcs);

BOOL AddActivity(Process\* pPrcs, Activity\* pActNew)

{

Activity\* pAct;

State\* pOutPrev;

int outPrevId;

pAct = (Activity\*)GetHead(pPrcs->\_activities);

if (!pAct) {//First activity in process

pOutPrev = pPrcs->\_initState;

outPrevId = GetStateId(pOutPrev);

}

else {

pAct = (Activity\*)GetTail(pPrcs->\_activities);

pOutPrev = pAct->\_outputState;

outPrevId = GetStateId(pOutPrev);

}

if (outPrevId != GetStateId(pActNew->\_inputState))

return FALSE;

return AddTail(pPrcs->\_activities, pActNew);

}

void SetInitialState(Process\* pPrcs, State\* pState);

State\* GetInitialState(Process\* pPrcs);

BOOL Execute(Process\* pPrcs);

State\* GetFinalState(Process\* pPrcs);

**פתרון שאלה 2 חלק א'**

**א.**

template<class T>

class List

{

public:

List();

virtual ~List();

T\* GetHead();

T\* GetNext();

bool AddTail(const T&);

T\* GetTail();

};

**ב.**

class State

{

protected:

int \_id;

public:

State(int id) : \_id(id) { };

virtual ~State();

int GetStateId();

};

class Activity

{

State\* \_pInputState;

State\* \_pOutState;

public:

Activity(State\* pIn, State\* pOut);

virtual ~Activity();

State\* GetInputState() const;

void SetInputState(const State\* inState);

State\* GetOutputState() const;

virtual bool Execute() = 0;

};

**ג.**

class Process

{

protected:

List<Activity> \_activities;

State\* \_pInitState;

State\* \_pFinalState;

public:

bool AddActivity(const Activity& act);

void SetInitialState(State\* initState);

State\* GetInitialState() const;

bool Execute();

State\* GetFinalState() const;

};

bool Process::Execute()

{

State\* pCurState = GetInitialState();

for(Activity\* pAct = \_activities.GetHead(); pAct; pAct = \_activities.GetNext()) {

pAct->SetInputState(pCurState);

if (!pAct->Execute())

return false;

pCurState = pAct->GetOutputState();

}

\_pFinalState = pCurState;

return true;

};

שאלת C++ חלק ב (20 נק') מה יודפס?

(1) ------

X::X()

(2) ------

X::X(X&)

(3) ------

X::X()

Y::Y()

(4) ------

X::X(X&)

X::X(X&)

Y::Y(X)

(5) ------

X::X(X&)

Y::Y(Y&)

(6) ------

X::X(X&)

X::X(X&)

X::X(X&)

Y::Y(X)

X::X(X&)

Z::Z(X)

(7) ------

X::X(X&)

Y::Y(Y&)

Z::~Z()

פתרון

BASH

א.

#!bin/bash

len=`echo $1|wc -c`

if [[ $len -lt 3 ]]; then echo $1; fi

ב.

#! /bin/bash

ccc=0

for i in $1/\*; do

if [[ -f "$i" ]] ; then

(( ccc=$ccc+1 ))

elif [[ -d "$i" ]]; then

a=`$0 $i`

(( ccc=$ccc + $a ))

fi

done

echo $ccc

שאלת UML + DP

א.

1. דיאגרמת מקלות, או ריבוע עם stereotype <<actor>>
2. כדי שלכולן יהיה ממשק משותף, והסובייקט יוכל להכיר רק ממשק אחד ולעבוד בעזרתו עם כל האובזרוורים.
3. כדי שגורם חיצוני לא יוכל ליצור אובייקטים נוספים..

ב

