1. teorie

Může být každá z odpovědí správná i špatná. 5 bodů pokud je vše správně

1. Vyberte správné tvrzení
	1. Stav objektu implementují metody
	2. Metoda v části private není viditelná mimo modul
	3. Objekt má stav, ale nemá chování
	4. Objekt neobsahuje vnitřní paměť
2. Vyberte špatné příklady dědičnosti
	1. Kružnice -> koule
	2. Úsečka -> přímka
	3. Geometrický útvar -> bod
	4. Obdélní -> čtverec
3. Vyberte nesprávné tvrzení
	1. Dědit v Borland pascalu lze od více objektů
	2. Potomek může dědit jen metody
	3. Dědičnost nelze použít skrz moduly
	4. Borland Pascal přímo podporuje abstraktní metody
4. Zapouzdření
	1. Způsobuje, že se k objektu nedostane nikdo jiný než vlastník
	2. Není implementováno v jazyce Borland Pascal
	3. Zakazuje přístup k některým metodám objektu
	4. Nelze charakterizovat heslem: „Nikdo nesmí mít šanci zjistit nic o tom, jak to dělám, že umím to, co umím.“
5. Vyberte správné tvrzení
	1. Zpráva umožňuje dostat se přes slupku zapouzdření
	2. Virtuálním metodám překladač vyhradí místo v paměti již v době překladu a také řeší všechny vztahy mezi metodami a objekty již v době překladu
	3. Direktiva virval označuje virtuální metodu
	4. Polymorfismus umožňuje, aby každý objekt ve stromové hierarchii objektů volal metodu se stejným jménem
6. Máme program na evidenci adres osob. Které z následujících metod nepatří do aplikační vrstvy
	1. Zápis nové osoby do souboru
	2. Ověření existence souboru
	3. Vyhledávání osob dle části adresy
	4. Zpracování dat z klávesnice
7. návrh struktur – 40 bodů

navrhněte objektové struktury pro program knihovna – povinné role uživatelů čtenář (rezervuje, prodlužuje výpůjčky, zobrazuje svoje výpůjčky), správce knihovny (nové knihy, kontroluje výpůjčky), anonymní uživatel (zobrazuje katalog knih). Čtenář a správce musí logovat. Návrh musí obsahovat UML, popis rozvrstvení a slovní popis klíčových metod.

1. opravy kódu
2. opravte tak, aby bylo vše správně. Doplňte metody. Tučné musí zůstat nezměněno (10bodů)

program Kresleni;

type Lokace = object

 X:integer;

 Y: integer;

 End;

 Bod= object(Lokace)

 Stav: boolean;

 Procedure Init;

 **Procedure Zobraz; virtual;**

Procedure Smaz;

 End;

 Kruh = object(Bod)

 Polomer: integer;

 Barva: word;

 Procedure Init;

 Procedure Zobraz;

Procedure Smaz;

 End;

1. Opravte tak, aby byl kód správný a funkční

(10 bodů)

unit complex;

INTERFACE

type CComplex=object(Complex\_abs)

private

 Re:Real;

 Im:Real;

public

 procedure init(R,I,:Real);

 function getRe:Real;

 function getIm:Real;

end;

type Complex\_abs=object

public

 function abs:Real;

end;

IMPLEMENTATION

procedure CComplex.init(R,I:Real);

begin

 Re:=R;IM:=I;

end;

function CComplex.getRe:Real;

begin

 getRe:=Re;

end;

function CComplex.getIm:Real;

begin

 getIm:=Im;

end;

function CComplex.abs:Real;

begin

 abs:=sqrt(Re\*Re+Im\*Im);

end;

var cislo:CComplex;

BEGIN

END.

uses complex;

procedure nasob(a,b,c,d,v:CComplex

begin

 v.init(a.getRe\*c.getRe-b.getRe\*d

b.getRe\*c.getRe+-a.getRe\*d.get

end;

var c1,c2,s:CComplex;

begin

 c1.Re:=3;

 c1.Im:=5;

 c2.init(2,3);

 nasob(c1,c2,s);

 writeln(‘Cislo je:’s.getRe,’+’s.get ???

 readln;

end.

1. Rozdělení do vrstev (10 bodů). Proveďte oddělení prezentační vrstvy u programu

uses crt, lifo;

var retezec, co:string

 i, cislo, x, y, err: integer;

 z: CLifo;

procedure vyhodnot(co:string);

var i, cislo, x, y, err:integer;

begin

 case co[1] of

 '+';

 Begin

 x:=z.pop;

 y:=z.pop;

 z.push(x+y);

 end;

 '-';

 Begin

 x:=z.pop;

 y:=z.pop;

 z.push(x-y);

 end;

 '\*';

 Begin

 x:=z.pop;

 y:=z.pop;

 z.push(x\*y);

 end;

 '/';

 Begin

 x:=z.pop;

 y:=z.pop;

 z.push(x div y);

 end;

 else begin

 val(co, coslo, err);

 z.push(cislo);

 end;

 end;

end;

Begin

 z.init;

 clrscr;

 write('retezec'); readln(retezec);

 writeln('------------------------------');

 repeat

 co:=copy(řetězec,1,pos(' ',retezec)-1);

 delete(řetězec,1,pos(' ',retezec));

 vyhodnotit(co);

 until pos(' ',retezec)=0;

 vyhodnotit(retezec);

 writeln(z.pop);

 readln;

end;

**Unit LIFO má následující hlavičku:**

unit lifo;

interface

type CLifo=object

 public

 procedure init;

 procedure push(i:integer);

 function pop:integer;

 function empty:boolean;

 function full:boolean;

 function isError:boolean;

 function getError:string;

 private

 Items:array[1..10] of integer;

 StackTop:integer;

 MaxItemsCount:integer;

 LastError:integer;

 End;