שאלון + פתרון

0660603202.

המסלול האקדמי המכללה למינהל

המסלול האקדמי המכללה למינהל

ביה"ס למדעי המחשב

ברקוד נבחן

ת.ז הסטודנט: _____

מספר חדר: _____

מספר נבחן: _____

מספר אסמכתא: _____

מבחן בקורס: תכנות מונחה עצמים

תאריך הבחינה: 13.11.04

שנת הלימודים: תשע"ג, **סמסטר:** קיץ, **מועד:** ב'

משך הבחינה: 3.5 שעות

שם המרצה/ים:

אליהו חלסצ'י

שם המתרגל/ים:

חיים שפיר

מבנה הבחינה: הבחינה מורכבת מחלק אחד.

מספר השאלות הכולל בבחינה: 6.

משקל כל שאלה: בצמוד לכל שאלה

הוראות לנבחן:

- אסור השימוש בכל חומר עזר
- יש לענות בגוף השאלון.
- נדרש להחזיר את השאלון.
- לא מצורף נספח לבחינה
- מחברת טיוטה: כן
- מחברת נפרדת לכל שאלה: לא
-

בהצלחה!!

שאלה 1 – Constructors and virtual methods (10 נקודות)

נתונות ההגדרות של המחלקות הבאות :

```
class A{
  public:
  A(){
     cout<<"A"<<endl;
     m();   // notice!!
  }
  void call_m(){
     //...
     m();
  }
  virtual void m(){
     cout<<"A::m()"<<endl;
  }
  ~A(){
     cout<<"~A"<<endl;
     m();   // notice!!
  }
};

class B: public A{
   public:
   B(){
      cout<<"B"<<endl;
   }
   virtual void m(){
      cout<<"B::m()"<<endl;
   }
   ~B(){
      cout<<"~B"<<endl;
   }
};

class C: public B{
   public:
   C(){
      cout<<"C"<<endl;
   }
   void m(){
      cout<<"C::m()"<<endl;
   }
   ~C(){
      cout<<"~C"<<endl;
   }
};

int main()
```

```
{
  C c;
  c.call_m();
  return 0;
}
```

מהו הפלט של התוכנית? (10 נק')

תשובה:

```
A
A::m()
B
C
C::m()
~C
~B
~A
A::m()
```

שאלה 2: ירושה ופולימורפיזם (30 נקודות)

אנו בונים סימולציה של משחק כדורעף. נתונה המחלקה הבאה המגדירה מהו שחקן.

```
class VBplayer{
   int x,y;  // current position
   char* name;
   int speed;
   double accuracy;
    // …
   public:
   VBplayer(// …){//…}
   // …
 };
```

לשחקנים בקבוצה, שישה במספר, יש תפקידים שונים. למשל השלושה האחוריים אחראים על קבלה (הגנה), ואילו הקדמיים אחראיים על התקפה – האמצעי מכין את הכדור להנחתה עבור המנחית הימני או השמאלי שמעביר את הכדור בחוזקה לצד השני של המגרש.

התחלנו לייצג את התפקידים הללו במחלקות שונות. כל תפקיד במחלקה משלו עם תכונות מתאימות.

```
class VBsetter{ // …};      // תפקיד המכין האמצעי
class VBhitter{// …};      // תפקיד המנחית (שמאלי או ימני)
class VBlibero{// …};      // שחקן קו אחרי
```

בתנאים מסוימים לאחר שהושגה נקודה לטובת הקבוצה מתבצע חילוף. השחקנים מחליפים עמדות עם כיוון השעון וגם תפקידים. למשל המנחית הימני עובר לקו אחורי. המכין (האמצעי) עובר לתפקיד המנחית הימני. המנחית השמאלי עובר להיות המכין. והשחקן השמאלי האחורי עובר להיות המנחית השמאלי וכו'.

עלינו לאפשר את החילופים הללו דינאמית כלומר בזמן ריצה. אם למשל יש לנו מופע של VBplayer (עם שם ותכונות משלו) ייתכן ובזמן ריצה נצטרך לשנות לו את התפקיד (לדוגמא מ VBsetter ל VBhitter)

א. ערכו את הקוד לעיל כך שיתמוך בביצוע חילופים (10 נק')

(רמז: חישבו על איזה data member יש להוסיף ל VBplayer)

ב. הוסיפו בונה ומתודות מתאימות, וצרו ב main מופע של שחקן כדורעף בשם "Eli" עם מהירות 5 ודיוק של 0.9 ותפקיד של VBsetter במיקום 4,8. בשורת הקוד הבאה שנו את תפקידו ל VBhitter ובשורה שאחריה את מיקומו ל 8,6 (10 נק')

ג. כל 12 השחקנים (משתי הקובצות) מוחזקים במערך מסוג **VBplayer. כתבו פונקציה המחזירה את הדיוק הגבוה ביותר של השחקנים שתפקידם הנוכחי הוא VBhitter (10 נק')

פתרון:

א. ניצור מחלקה בשם VBtask שמגדירה תפקיד של שחקן כדורעף. כל מחלקות התפקידים יירשו אותה וכך ניצור מכנה משותף בין כולן. במחלקה VBplayer ניצור data member מסוג *VBtask ונוסיף מתודות set ו get מתאימות כדי לאפשר את חלופת התפקיד בעת הצורך.

ב. נוסיף בונה שמאתחל את כל שדות המחלקה של VBplayer (כולל ה VBtask) וכן מתודות set עבור המיקום. (מתודת set עבור ה VBtask יצרנו בסעיף הקודם כדי לאפשר חילוף). כעת ב main נוכל ליצור מופע של שחקן כדורעף ולהדביק לו מופע דינאמי של VBsetter ובשורה הבאה לשנות לו את התפקיד ל VBhitter ואת המיקום בהתאמה.

ג. ל VBplayer נוודא שיש getter ל VBtask ול accuracy. ובפונקציה כשנעבור על המערך נבדוק באמצעות typeid האם הטיפוס של תפקיד השחקן הנוכחי מתאים ל VBhitter ואם כן נשמור את הדיוק המקסימאלי ונחזיר אותו בסוף.

קוד:

```
class VBtask{};
class VBsetter : public VBtask{};
class VBhitter : public VBtask{};
class VBlibero : public VBtask{};


class VBplayer{
    int x,y;
    char* name;
    int speed;
    double accuracy;
```

```
VBtask* task;
public:
VBplayer(const char* a_name, int a_speed, double a_accuracy, VBtask* a_task,int ax,int ay){
   name=new char[strlen(a_name)+1];
   strcpy(name,a_name);
   speed = a_speed;
   a_accuracy = accuracy;
   task = a_task;
   x=ax;
   y=ay;
 }
 void setTask(VBtask* a_task){
    task=task;
 }
 const VBtask* getTask(){
    return task;
 }
 void setPosition(int ax,int ay){
    x=ax;
    y=ay;
 }
 double getAccuracy(){
    return accuracy;
 }
};

double getMaxAccuracy(VBplayer ** players){
   double max=0;
   for(int i=0;i<12;i++){
      if (typeid(players[i]->getTask())==typeid(VBhitter) && max<players[i]->getAccuracy())
         max=players[i]->getAccuracy();
   }
   return max;
 }
```

```
int main()
{
    VBplayer eli("eli",5,0.9, new VBsetter(), 4,8);
    eli.setTask(new VBhitter());
    eli.setPosition(8,6);
    return 0;
}
```

שאלה 3: Operator Overloading – Const (20 נקודות)

נתונה המחלקה Student שאינה ניתנת לעריכה:

```
class Student{
  char* name;
  public:
  Student(const char* a_name){
    name=new char[strlen(a_name)+1];
    strcpy(name,a_name);
  }
  const char* getName() const {return name;}
};
```

נתונה המחלקה הבאה המכילה מערך של סטודנטים

```
class MyStudentArray{
  Student** arr;
  int size;
   // ...
}
```

כמו כן, נתונה פונקציית ה main הבאה:

```
int main() {
  MyStudentArray a(3);  // create an inner array of 3 Students
  a[0]= new Student("a");
  a[1]= new Student("b");
  a[2]= new Student("c");
  cout<<a[0]<<endl;
  cout<<a<<"---"<<endl;
  return 0;
}
```

פלט:

```
a
a
b
c
---
```

השלימו את הקוד החסר כך שפונקציית ה main תעבוד ללא בעיות. שימו לב שה main לא בודקת את כל מקרי הקצה, ואתם נדרשים להקפיד על יעילות, צורת העברת הפרמטרים השונים (בדגש על const) וערכי החזרה.

תשובה:

```
class MyStudentArray{
  Student** arr;
  int size;
  public:
  MyStudentArray(int a_size){
     size=a_size;
     arr=new Student*[size];
  }
  int getSize() const { return size;}
  Student*& operator[] (int index) { return arr[index];}
  const Student* operator[](int index) const { return arr[index];}
};

ostream& operator<<(ostream& out,const Student* s){
   out<<s->getName()<<endl;
   return out;
}

ostream& operator<<(ostream& out,const MyStudentArray& a){
   for(int i=0;i<a.getSize();i++){
      out<<a[i]<<endl;
   }
   return out;
}
```

שאלה 4 – Templetes (20 נקודות)

נתונה פונקציית ה main הבאה:

```
int main() {
  int i=2;
  float f=2.2;
  Trip trip;            // notice!!!
  trip(i);
  trip(f);
  cout << i << endl;  // 6
  cout << f << endl;  // 6.6
  return 0;
}
```

למרות ש i ו f הינם טיפוסים שונים, הפעולה trip משלשת את ערכם

א. ממשו את הנדרש כדי שהקוד ב main יעבוד. (10 נק')

ב. כתבו פונקציה בשם Apply הפועלת על מבנה נתונים לא ידוע (יכול להיות מערך, רשימה מקושרת, עץ וכו'), המכיל נתונים מסוג לא ידוע (מכיל למשל סטודנטים או עובדים או int ים וכו') ובנוסף הפונקציה תקבל כפרמטר פעולה כלשהי. הפונקציה Apply תפעיל את הפעולה הנ"ל על כל אחד מהנתונים במבנה הנתונים. (10 נק')

תשובה:

א.

```
class Trip{
  public:
  template <class T>
  void operator()(T& t){
    t*=3;
  }
};
```

ב.

```
template<class iterator, class operation>
void Apply(iterator begin, iterator end, const operation& f) {
for( ; begin != end ; begin++)
```

```
    f(*begin);

}
```

שאלה 5 שאלות פתוחות (10 נק')

א. בשאלה 4 בסעיף א', מהו Trip מהו trip? (4 נק')

ב. בשאלה 4 בסעיף ב', מהם ההנחות הסמויות שהנחת על הפרמטרים השונים? (4 נק')

ג. מה הקשר בין Trip ל Apply? (2 נק')

תשובות:

א. Trip היא מחלקה המגדירה object function – מחלקה עם אופרטור () הפועל על טיפוס T כלשהו ומכפיל את ערכו פי 3. trip הוא אובייקט שנוצר ממחלק הזו ובאמצעותו ניתן להפעיל את האופרטור ().

ב. שלאובייקט מסוג iterator מומשו האופרטורים =! , ++ , *, ושניתן לקבל אותם by value בבטיחות. ולאובייקט מסוג operation קיים האופרטור () הפועל על טיפוס פרמטרי T.

ג. אובייקט מסוג Trip ניתן להעברה כפרמטר הפעולה עבור apply, אנו מעבירים "פונקציה" כפרמטר אבל בפעול פשוט מעבירים כפרמטר אובייקט של Dub.

שאלה 6: ירושה מרובה (10 נקודות)

האם קטע הקוד הבא מתקמפל, אם לא – מדוע?

אם כן – מה מודפס למסך?

```
class Shape{
   char color;
   public:
   Shape(char c){
      color=c;
      cout<<"Shape color "<<c<<endl;
   }
   char getColor(){return color;}
   // ...
};

class Rectangle : virtual public Shape{
   int width,height;
   public:
   Rectangle(char c):Shape(c){
      cout<<"Rectangle color "<<c<<endl;
   };
   //...

};
```

```
class Circle : virtual public Shape{
   int x,y,r;
   public:
   Circle(char c):Shape(c){
      cout<<"Circle color "<<c<<endl;
   };
   //...
};

class Cylinder : public Rectangle, public Circle{
   public:        // 'r'                          //'s'          // 't'
   Cylinder(char c):Shape(c),Rectangle(c+1),Circle(c+2){};
};

int main()
{
  Cylinder c('r');
  cout<<c.getColor()<<endl;
}
```

תשובה : הקוד מתקמפל.

פלט:

```
Shape color r
Rectangle color s
Circle color t
r
```

בהצלחה