PROGRAMI I ORIENTUAR NE OBJEKTE

Polimorphism

DETYRA 1

PRISHTINË

```
#include <iostream>
using namespace std;

class Shape {
   protected:
      int width, height;
   public:
      Shape( int a=0, int b=0)
      {
         width = a;
         height = b;
      }
      int area()
      {
         cout << "Parent class area :" <<endl;
         return 0;
      }
};
class Rectangle: public Shape{
   public:
      Rectangle( int a=0, int b=0):Shape(a, b) { }
      int area ()
      {
         cout << "Rectangle class area :" <<endl;
         return (width * height);
      }
};
```

```
28 class Triangle: public Shape{
29    public:
30       Triangle( int a=0, int b=0):Shape(a, b) { }
31       int area ()
32       {
33          cout << "Triangle class area :" <<endl;
34          return (width * height / 2);
35       }
36 };
37  // Main function for the program
38 int main( )
39  {
40     Shape *shape;
41     Rectangle rec(10,7);
42     Triangle  tri(10,5);
43
44     // store the address of Rectangle
45     shape = &rec;
46     // call rectangle area.
47     shape->area();
48     cout<< rec.area()    << '\n';
49     // store the address of Triangle
50     shape = &tri;
51     // call triangle area.
52     shape->area();
53     cout<<tri.area() << '\n';
54     cin.get();
55     return 0;
56 }
```

PURE VIRTUAL FUNCTIONS:


```
19  class Shape {
20      protected:
21          int width, height;
22      public:
23          Shape( int a=0, int b=0)
24          {
25              width = a;
26              height = b;
27          }
28          // pure virtual function
29          virtual int area() = 0;
30  };
```

DETYRA 2

PRISHTINË

```
1  #include <iostream>
2  using namespace std;
3
4  class CPolygon
5  {
6      protected:
7          int width, height;
8      public:
9          void setup (int first, int second)
10         {
11             width= first;
12             height= second;
13         }
14 };
15
16 class CRectangle: public CPolygon
17 {
18     public:
19         int area()
20         {
21             return (width * height);
22         }
23 };
24
25 class CTriangle: public CPolygon
26 {
27     public:
28         int area()
29         {
30             return (width * height / 2);
31         }
32 };
```


```
34  int main ()
35  {
36      CRectangle rectangle;
37      CTriangle triangle;
38
39      CPolygon * ptr_polygon1 = &rectangle;
40      CPolygon * ptr_polygon2 = &triangle;
41
42      ptr_polygon1->setup(2,2);
43      ptr_polygon2->setup(2,2);
44
45      cout << rectangle.area () << endl;
46      cout << triangle.area () << endl;
47      cin.get();
48      return 0;
49  }
```

DETYRA 3: VIRTUAL MEMBER


```cpp
#include <iostream>
using namespace std;

class CPolygon
{
    protected:
        int width, height;
    public:
        void setup (int first, int second)
        {
            width= first;
            height= second;
        }
        virtual int area()
        {
            return (0);
        }
};

class CRectangle: public CPolygon
{
    public:
        int area()
        {
            return (width * height);
        }
};
```

```
29  class CTriangle: public CPolygon
30  {
31      public:
32          int area()
33          {
34              return (width * height / 2);
35          }
36  };
37
38  int main ()
39  {
40      CRectangle rectangle;
41      CTriangle triangle;
42      CPolygon polygon;
43
44      CPolygon * ptr_polygon1 = &rectangle;
45      CPolygon * ptr_polygon2 = &triangle;
46      CPolygon * ptr_polygon3 = &polygon;
47
48      ptr_polygon1->setup(2,2);
49      ptr_polygon2->setup(2,2);
50      ptr_polygon3->setup(2,2);
51
52      cout << ptr_polygon1->area () << endl;
53      cout << ptr_polygon2->area () << endl;
54      cout << ptr_polygon3->area () << endl;
55
56      cin.get();
57      return 0;
58  }
```

DETYRA 4: DYNAMIC ALLOCATION

PRISHTINË

```
1  #include <iostream>
2  using namespace std;
3
4  class CPolygon
5  {
6      protected:
7          int width, height;
8      public:
9          void setup (int first, int second)
10         {
11             width= first;
12             height= second;
13         }
14         virtual int area(void) = 0;
15         void onscreen(void)
16         {
17             cout << this->area() << endl;
18         }
19 };
20
21 class CRectangle: public CPolygon
22 {
23     public:
24         int area(void)
25         {
26             return (width * height);
27         }
28 };
```

```
30  class CTriangle: public CPolygon
31  {
32      public:
33          int area(void)
34          {
35              return (width * height / 2);
36          }
37  };
38
39  int main ()
40  {
41      CPolygon * ptr_polygon1 = new CRectangle;
42      CPolygon * ptr_polygon2 = new CTriangle;
43
44      ptr_polygon1->setup(2,2);
45      ptr_polygon2->setup(2,2);
46
47      ptr_polygon1->onscreen();
48      ptr_polygon2->onscreen();
49
50      delete ptr_polygon1;
51      delete ptr_polygon2;
52
53      cin.get();
54      return 0;
55  }
```

PRISHTINË