Условие:

Создать абстрактный базовый класс Figure с виртуальной функцией - Периметр. На его основе реализовать производные классы Rectangle, Circle, в которых данная функция переопределена. В функции main определить массив указателей на абстрактный класс, в котором присваиваются адреса различных объектов.

Исходный код программы:  
#include **<iostream>  
  
using namespace** std;  
  
**class** Figure {  
**public**:  
 **virtual double** getSquare() { **return** 0; };  
 **virtual double** getPerimeter() { **return** 0; };  
 **virtual void** showFigureType(){};  
};  
**class** Rectangle : **public** Figure {  
**private**:  
 **double** width;  
 **double** height;  
  
**public**:  
 Rectangle(**double** w, **double** h)  
 : width(w)  
 , height(h)  
 {  
 }  
 **double** getSquare()  
 {  
 **return** width \* height;  
 }  
 **double** getPerimeter()  
 {  
 **return** width \* 2 + height \* 2;  
 }  
 **void** showFigureType()  
 {  
 cout << **"Rectangle"** << **"\n"**;  
 }  
};  
**class** Circle : **public** Figure {  
**private**:  
 **double** radius;  
  
**public**:  
 Circle(**double** r)  
 : radius(r){};  
  
 **double** getSquare()  
 {  
 **return** radius \* radius \* 3.14;  
 }  
 **double** getPerimeter()  
 {  
 **return** 2 \* 3.14 \* radius;  
 }  
 **void** showFigureType()  
 {  
 cout << **"Circle"** << **"\n"**;  
 }  
};  
  
**int** main()  
{  
 Rectangle\* rec[4];  
 rec[0] = **new** Rectangle(30, 50);  
 rec[1] = **new** Rectangle(29, 50);  
 rec[2] = **new** Rectangle(28, 50);  
 rec[3] = **new** Rectangle(27, 50);  
 **for** (**int** i = 0; i < 4; i++) {  
 cout << **"Rectangle square: "** << rec[i]->getSquare() << **"\n"**;  
 }  
  
 Circle\* circle[3];  
 circle[0] = **new** Circle(25);  
 circle[1] = **new** Circle(25);  
 circle[2] = **new** Circle(25);  
 **for** (**int** i = 0; i < 3; i++) {  
 cout << **"Circle square: "** << circle[i]->getSquare() << **"\n"**;  
 }  
  
 **return** 0;  
}

Скриншоты программы:

