

Zadanie 1

Zamień składowe typu metody na składowe typu property

```
#pragma once
using namespace System;

ref class Prostokat
{ protected: double wymiar;
  protected: double dlugosc;

public:
    Prostokat (double szer, double dl): wymiar(szer),dlugosc(dl) {}

    Prostokat (): wymiar(1.0),dlugosc(1.0) {}

    String^ info()
        { return L"To jest prostokat. "; }

    double Pole()
        { return wymiar*dlugosc; }

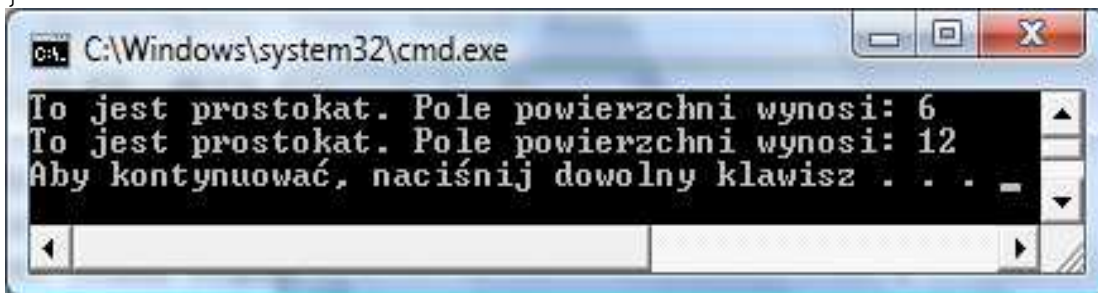
    virtual String^ ToString() override
        { return info() +L"Pole powierzchni wynosi: " + Pole(); }

};

// Figury0_.cpp : main project file.

#include "stdafx.h"
#include "figurykol0_.h"
using namespace System;

int main(array<System::String ^> ^args)
{
    array<Prostokat^>^ figury = gcnew array<Prostokat^>(2);
    figury[0] = gcnew Prostokat(2.0, 3.0);
    figury[1] = gcnew Prostokat(3.0, 4.0);
    for (int i=0; i<figury->Length; i++)
        Console::WriteLine(figury[i]->ToString());
    return 0;
}
```



Rys. 1

Rozwiązanie zadania 1

```
#pragma once
using namespace System;

ref class Prostokat
{ protected: double wymiar;
protected: double dlugosc;
public:
    Prostokat (double szer, double dl): wymiar(szer),dlugosc(dl) {}

    Prostokat (): wymiar(1.0),dlugosc(1.0) {}

    property String^ info
    { String^ get()
        { return L"To jest prostokat. ";}
    }

    property double Pole
    { double get()
        { return wymiar*dlugosc; }
    }

    virtual String^ ToString() override
    { return info +L"Pole powierzchni wynosi: " + Pole; }
};
```

Zadanie 2

Uzupełnij brakujące metody, aby program główny działał poprawnie (rys.2)

```
#pragma once
using namespace System;

ref class Kolo
{
    protected:    double wymiar;
public:    Kolo (double pr): wymiar(pr) {}
           Kolo (): wymiar(1.0) {}

           virtual property String^ info
           {
               String^ get()
               {
                   return L"To jest kolo. ";
               }
           }

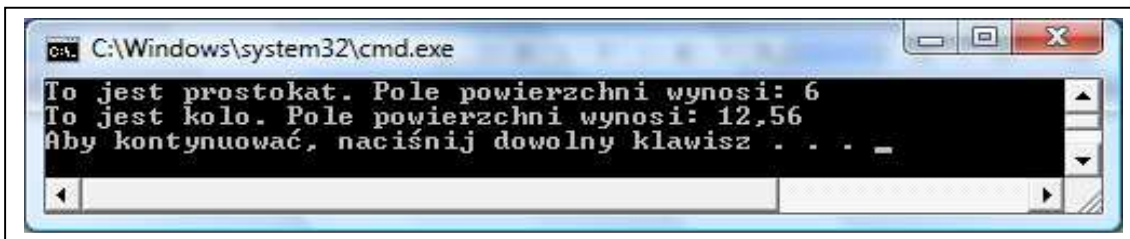
           virtual double Pole()
           {
               return 3.14*wymiar*wymiar;
           }

           virtual String^ ToString() override
           {
               //uzupełnić ciało metody
           };
};

ref class Prostokat : Kolo
{
    protected: double dlugosc;
public: Prostokat (double szer, double dl): Kolo(szer),dlugosc(dl) {}
           Prostokat (): Kolo(1.0),dlugosc(1.0) {}
           //uzupełnić metody
};

};
/ Figury1.cpp : main project file.
#include "stdafx.h"
#include "figurykoll1.h"
using namespace System;

int main(array<System::String ^> ^args)
{
    array<Kolo^>^ figury = gcnew array<Kolo^>(2);
    figury[0] = gcnew Prostokat(2.0, 3.0);
    figury[1] = gcnew Kolo(2.0);
    for (int i=0; i<figury->Length; i++)
        Console::WriteLine(figury[i]->ToString());
    return 0;
}
```



Rys. 2

Rozwiązanie zadania 2

```
#pragma once
using namespace System;

ref class Kolo
{
protected:
    double wymiar;
public:
    Kolo (double pr): wymiar(pr) {}
    Kolo (): wymiar(1.0) {}
    virtual property String^ info
    {
        String^ get()
        {
            return L"To jest kolo. ";
        }
    }
    virtual double Pole()
    {
        return 3.14*wymiar*wymiar;
    }

    virtual String^ ToString() override
    {
        return info +L"Pole powierzchni wynosi: " + Pole();
    }
};

ref class Prostokat : Kolo
{
protected:
    double dlugosc;
public:
    Prostokat (double szer, double dl): Kolo(szer),dlugosc(dl) {}
    Prostokat (): Kolo(1.0),dlugosc(1.0) {}
    virtual property String^ info
    {
        String^ get()override
        {
            return L"To jest prostokat. ";
        }
    }
    virtual double Pole() override
    {
        return wymiar*dlugosc;
    }
};

// Figury1.cpp : main project file.
#include "stdafx.h"
#include "figurykoll1.h"
using namespace System;

int main(array<System::String ^> ^args)
{
    array<Kolo^>^ figury = gcnew array<Kolo^>(2);
    figury[0] = gcnew Prostokat(2.0, 3.0);
    figury[1] = gcnew Kolo(2.0);
    for (int i=0; i<figury->Length; i++)
        Console::WriteLine(figury[i]->ToString());
    return 0;
}
```