Zaawansowane programowanie w języku C++
Programowanie obiektowe

Prezentacja jest współfinansowana przez
Unię Europejską w ramach
Europejskiego Funduszu Społecznego w projekcie pt.

„Innowacyjna dydaktyka bez ograniczeń - zintegrowany rozwój Politechniki Łódzkiej - zarządzanie Uczelnią, nowoczesna oferta edukacyjna i wzmacniania zdolności do zatrudniania osób niepełnosprawnych”

Prezentacja dystrybuowana jest bezpłatnie

Programowanie obiektowe

- Naturalny sposób postrzegania świata (brak konieczności podziału programu na dane i kod – logikę)
- Możliwość ponownego wykorzystania kodu
- Łatwe rozszerzanie i dostosowywanie funkcjonalności

- Cechy programowania obiektowego:
  - Abstrakcja
  - Enkapsulacja
  - Dziedziczenie
  - Polimorfizm

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Zadanie

Oprogramowanie biblioteki pozwalającej na budowanie nowych figur geometrycznych oraz wykorzystanie już istniejących:

- Modele: figura, prostokąt, kwadrat, koło …
- Identyczny sposób użycia niezależnie od rodzaju figury
- Zapewnienie wspólnego interfejsu

Należy odpowiedzieć na pytania:

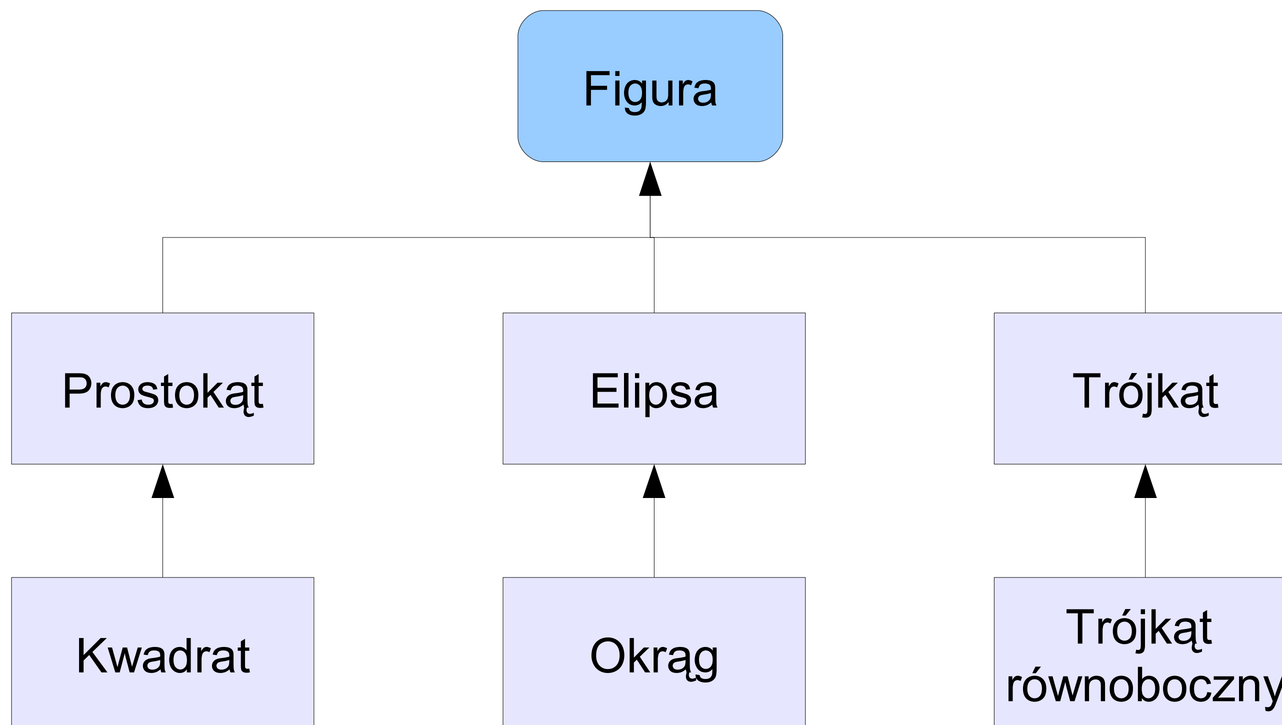
- Jakie cechy wspólne mają figury geometryczne?
- Czy występuje jakaś hierarchia figur i można określić zależności między nimi?

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Przykładowa hierarchia obiektów



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Klasa Figura

```
class Figura
{
    public:
        double Pole() const { return 0.0; }
        double Obwod() const { return 0.0;}
};
```

- Metody Pole i Obwod są pojęciami abstrakcyjnymi
- Co jest instancją klasy Figura?

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Klasa Prostokąt

```
class Prostokat
{
    private:
        double a, b;
    public:
        Prostokat() : a( 0 ), b( 0 ) {}
        Prostokat( double _a, double _b )
                        : a( _a ), b( _b ) {}
        double Pole() const { return a*b; }
        double Obwod() const { return 2*a + 2*b; }
};
```


KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Dziedziczenie

```
class Prostokat : public Figura
{
    private:
        double a, b;
    public:
        Prostokat() : a( 0 ), b( 0 ) {}
        Prostokat( double _a, double _b )
                        : a( _a ), b( _b ) {}

};
```

- Czy to działa?

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Dziedziczenie

```
class Prostokat : public Figura
{
    private:
        double a, b;
    public:
        Prostokat() : a( 0 ), b( 0 ) {}
        Prostokat( double _a, double _b )
                        : a( _a ), b( _b ) {}
        double Pole() const { return a*b; }
        double Obwod() const { return 2*a + 2*b; }
};
```


KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Kontrola dostępu do składowych

- Składowe klas możemy udostępniać lub nie:
  - private
  - protected
  - public
- Analogiczne kwantyfikatory używane są przy dziedziczeniu

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Reguły dziedziczenia

- Zanim wykona się konstruktor klasy pochodnej to wykonywane są konstruktory (zgodnie z hierarchią) klas bazowych
- Metody, które nie są dziedziczone:
  - Konstruktory
  - Destruktory
  - Operator przypisania (=)

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Cel naszego zadania ...

Korzystać w taki sam sposób z obiektów pochodzących od klasy Figura!

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Wykorzystanie hierarchii klas

```
void drukujPole( const Figura& fig )
{
    cout << fig.Pole() << endl;
}

int main()
{
    Figura fig;
    Prostokat p( 3, 4 );
    drukujPole( p );
    return 0;
}
```


KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Wykorzystanie hierarchii klas – metody wirtualne

- Aby wykorzystać dobrodziejstwo hierarchii klas należy:
  - Zdefiniować metody wirtualne
  - Obiekty przekazywać przez wskaźnik lub referencję

- Poprawmy tak, aby działało ...

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Klasa Figura – metody wirtualne

```
class Figura
{
    public:
        virtual double Pole() const { return 0.0; }
        virtual double Obwod() const { return 0.0;}
};
```

- Słowa kluczowe 'virtual' wystarczy użyć tylko raz dla danej metody w klasie bazowej.
- Użycie jej w klasach potomnych jest opcjonalne.

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego
UNIA EUROPEJSKA
EUROPEJSKI FUNDUSZ SPOŁECZNY

Wirtualne destruktory

- Złota zasada: destruktor w klasie bazowej powinien być zawsze wirtualny!
- Jeżeli destruktor nie jest wirtualny to podlega wczesnemu wiązaniu (statycznemu) przez kompilator

```
Figura *fig = new Prostokat( 2.4, 3.7 );
fig->Obwod();

delete fig;
```


KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Cel naszego zadania ...

Zapewnienie wspólnego interfejsu!

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Klasa Figura – klasa abstrakcyjna

```
class Figura
{
    public:
        virtual double Pole() const = 0;
        virtual double Obwod() const = 0;
};
```

- Metoda abstrakcyjna musi być przedefiniowana w klasie pochodnej
- Nie da się utworzyć instancji klasy, która zawiera co najmniej jedną metodę czysto wirtualną (abstrakcyjną)

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Dziedziczenie wielokrotne

```
class Wspolrzedna
{
    protected:
        double x, y;
    public:
        Wspolrzedna() : x( 0 ), y( 0 ) {}
        Wspolrzedna( double _x, double _y )
                    : x( _x ), y( _y ) {}
};

class Prostokat : public Figura, public Wspolrzedna
{
      …
};
```


KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Rozmieszczenie obiektów w pamięci

- Informacje o metodach wirtualnych przechowywane są w tzw. tablicach metod wirtualnych:
  - Virtual Method Table
  - Virtual Function Table
  - VTable

```
g++ fig.cc -Wall -pedantic -fdump-class-hierarchy

cat fig.cc.class
```


KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Dynamiczna kontrola typów

- Operator dynamic_cast:
  - Konwertuje wskaźnik klasy bazowej na wskaźnik do klasy pochodnej
  - Konwersja może się nie udać!
  - Należy zawsze sprawdzać czy nie zwrócił wartości NULL
- Operator typeid:
  - Wymaga włączenia pliku nagłówkowego <typeinfo>
  - Zwraca obiekt klasy std::type_info:
    - name()
    - before()
  - Wynik operacji może być porównywany (==) i (!=)

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Zaawansowane programowanie w języku C++
Przeciążanie operatorów

Prezentacja jest współfinansowana przez
Unię Europejską w ramach
Europejskiego Funduszu Społecznego w projekcie pt.

„Innowacyjna dydaktyka bez ograniczeń - zintegrowany rozwój Politechniki Łódzkiej - zarządzanie Uczelnią, nowoczesna oferta edukacyjna i wzmacniania zdolności do zatrudniania osób niepełnosprawnych”

Prezentacja dystrybuowana jest bezpłatnie