KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

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

# Operatory w języku C++

- Operator przypisania (=)
- Operatory arytmetyczne (+, -, *, /, %)
- Operatory mieszane (+=, -=, /= …)
- Operatory pre i post-inkrementacji/dekrementacji (++, –)
- Operatory porównania (==, !=, <, >, <=, >=)
- Operatory logiczne (!, &&, ||)
- Operator warunkowy (?)
- Operator ','
- Operatory bitowe ( &, |, ^, ~, <<, >> )
- Operator zasięgu '::'
- Operatory rzutowania
- Operatory new, delete, sizeof



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

# Przeciążanie operatorów

- Operatory możemy przeciążać podobnie jak inne funkcje i metody języka C++
- Operator się przeciąża poprzez zdefiniowanie funkcji operatorowej
- Argumentami operatorów są typami zadeklarowanymi zewnętrznie: struct, class, union, enum

```
typ_zwracany operator<op>( argumenty )
{
        …
}
```



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

# Przeciążanie operatorów

Przeciążanie operatorów nie jest dla programistów klas (bibliotek) ale dla użytkowników klas (bibliotek)!



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

# Przykład

```
class Num { int val; };

Num operator^( Num a, unsigned int b )
{
      Num ret; ret.val = 1
      for( unsigned int i = 0; i < b; ++i )
      {
            ret.val *= a.val;
      }
      return ret;
}

Num test; test.val = 2; test = test ^ 4;
```



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

# Przykład

```
class Num
{
   public:
       int val;
       Num& operator^( unsigned int b )
       {
              int val = 1
              for( unsigned int i = 0; i < b; ++i )
              {
                     val *= this->val;
              }
              this->val = val;
              return *this;
       }
};

Num test; test.val = 2; test ^ 4;
```



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

# Reguły przeciążania operatorów

- Zachowanie przedefiniowanego operatora nie musi mieć takiego samego znaczenia jak operatora pierwotnie zdefiniowanego
- W C++ nie można definiować nowych operatorów
- Nie wszystkie operatory da się przeciążyć:
  - . (kropka)
  - .*
  - ?:
  - ::
  - sizeof



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

# Reguły przeciążania operatorów - c.d.

- Operator przypisania (=)może być zdefiniowany tylko jako metoda
- Operatory wyłuskania spod wskaźnika (->) i (->*) mogą zwracać tylko i wyłącznie wskaźnik języka C++
- Operator tablicowy ([ ]) może przyjmować tylko jeden argument



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

# Operator tablicowy ([ ])

```
element& operator[]( unsigned int index )
{
      return tab[index];
}


const element& operator[]( unsigned int index ) const
{
      return tab[index];
}
```



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

# Operator przypisania (=)

```
class Num
{
   public:
       int val;
       Num& operator=( const Num& b )
       {
              this->val = b.val;
              return *this;
       }
};
```



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

# Operator przypisania (=) - c.d.

```
class Num
{
   public:
       int val;
       Num& operator=( const Num& b )
       {
              if ( this != &b )
                     this->val = b.val;
              return *this;
       }
};
```



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

# Operator przypisania (=) - c.d.

- Operator przypisania, a konstruktor kopiujący

```
Obj a = 2;
Obj b( a );
Obj c = a;
a = c;
...
```



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

# Operatory inkrementacji i dekrementacji

```
class Num
{
   public:
       int val;
       Num& operator++()
       {
              ++( this->val );
              return *this;
       }
       Num operator++( int x )
       {
              Num tmp; tmp.val = this->val;
              ++( *this );
              return tmp;
       }
};
```



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

# Operator funkcyjny ()

```
class Num
{
   public:
       int val;
       void someFunction( int arg );
       void operator()( int arg )
       {
              this->someFunction( arg );
       }
};

Num a; a.someFunction( 2 ); a( 2 );
```



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

# Operatory rzutowania

```
class Num;
{
   public:
       int val;
       operator int() const
       {
              return this->val;
       }
};

Num a; a.val = 2;
int b = a;
```



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

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