KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Zaawansowane programowanie w języku C++
Funkcje uogólnione - wzorce

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

Wzorce

- Cel: możliwość pisania kodu niezależnego od typów danych
- Środki:
  - Definicje makro preprocesora
  - Dziedziczenie i polimorfizm
  - Wzorce

Wzorce są to funkcje lub klasy napisane w sposób pozwalający na ich użycie z niezdefiniowanymi w momencie pisania typami (tzw. programowanie uogólnione).

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Warsztat – funkcja min

- Cel: napisanie funkcji zwracającej najmniejszy argument działającej z dowolnym typem, np:

  `min( 1, 3 ), min( 2.0, -1 ), min( "jeden”, "dwa" ), …`

- Algorytm:

```
if ( arg1 < arg2 )
    return arg1;
else
    return arg2;
```


KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Makro definicja

```
#define min(arg1, arg2) ( (arg1 < arg2) ? arg1 : arg2 )

int a = 2, b = 5;
int m = min( a, b );
```

- Zalety:
  - …
- Wady:
  - …

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Dygresja: słowo kluczowe inline

- Słowo inline służy do wskazania kompilatorowi funkcji lub metod, które chcielibyśmy rozwijać w miejscu wywołania (analogia do makro definicji)
- Inline jest tylko wskazówką dla kompilatora, który sam podejmuje decyzję, czy dana funkcja będzie rozwijana czy nie
- Funkcje i metody typu inline należy definiować w pliku nagłówkowym

- Zastosowanie:
  - ...

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Funkcja min – wersja dla typu int

```
inline int min( const int arg1, const int arg2)
{
    return (arg1 < arg2) ? arg1 : arg2;
}

int a = 2, b = 5;
int m = min( a, b );
```

- Zalety:
  - …
- Wady:
  - …

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Uogólniona wersja funkcji min

```
template< class T >
inline const T min( const T arg1, const T arg2)
{
    return (arg1 < arg2) ? arg1 : arg2;
}

int a = 2, b = 5;
int m = min( a, b );
```

- Zalety:
  - …
- Wady:
  - …

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Lepsza wersja uogólnionej funkcji min

```
template< class T >
inline const T& min( const T& arg1, const T& arg2)
{
    return (arg1 < arg2) ? arg1 : arg2;
}

int a = 2, b = 5;
int m = min( a, b );
```

- Zalety:
  - …
- Wady:
  - …

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Różne wywołania funkcji min

```
min( 4, 1 );
min( 4.0, 1 );
min< float >( 4, 1 );
```

```
min( "jeden", "dwa" );
min<std::string>( "jeden", "dwa" );
```

Czy to działa?

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Specjalizacja wzorca

- Często zachodzi potrzeba zdefiniowania specyficznego zachowania funkcji uogólnionej dla danego typu:
  - Dany typ ma inny interfejs
  - Dany typ ma inne znaczenie logiczne
  - Optymalizacja (np. std::vector<bool>)
- W takich przypadkach należy „przepisać” ponownie funkcję uogólnioną dla konkretnego typu danych.
- Kompilator wszędzie tam gdzie wystąpią podane przez nas w specjalizacji funkcji typy danych użyje wersji wyspecjalizowanej, a nie ogólnej funkcji

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Specjalizacja funkcji min dla typu std::string

```
template< class T >
inline const T& min( const T& arg1, const T& arg2 )
{
    return (arg1 < arg2) ? arg1 : arg2;
}

template<>
inline const std::string& min( const std::string& arg1,
                               const std::string& arg2 )
{
    return ( arg1.size() < arg2.size() ) ? arg1 : arg2;
}
```


KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Warsztat

- Wzorce, a wielkość pliku wynikowego
- Wzorce, a czas trwania kompilacji

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Parametry wzorców inne niż typ

- Możliwe jest przekazanie do wzorca danej funkcji nie typu lecz wartości
- Przykład:

```
template<size_t size>
inline void xchg( char *src, char *dst )
{
    char buffer[size];
    ...
    return;
}
```


KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Domyślne parametry wzorców

- Wzorzec funkcji może mieć wśród listy swoich parametrów parametry domyślne
- Parametry domyślne zachowują się identycznie jak domyślne argumenty funkcji.
- Przykład:

```
template<size_t size, class T = char*>
inline void xchg( T src, T dst )
{
    char buffer[size];
    ...
    return;
}
```


KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Klasy wzorcowe

- Język C++ umożliwia parametryzowanie nie tylko funkcji ale też klas, dzięki czemu można projektować kontenery niezależne od przechowywanych typów danych
- Najprostszym przykładem wzorca klasy jest parametryzowana struktura:

```
template< class T >
struct ListNode
{
    T node;
    ListNode<T> *next;
    ListNode<T> *prev;
};
```


KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Wzorzec klasy kontenera

```
template< class T >
class Container
{
    private:
        T value;
    public:
        Container() : T() {}
        Container( T& copy ) : T( copy ) {}
        Container( const Container<T>& copy ) :
                T( copy.value ) {}
        ...
};
```


KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Wzorce - podsumowanie

- Zalety:
  - Możliwość definiowania funkcji i klas niezależnych od typów danych, które przetwarzają lub przechowują
  - Bezpieczeństwo typów!
  - Wielokrotne wykorzystanie kodu
  - Możliwość tworzenia wszechstronnych bibliotek (biblioteka standardowa języka C++)
- Wady:
  - Zwiększony rozmiar aplikacji po skompilowaniu
  - Dłuższa kompilacja
  - Wzorce nie zawsze są dostępne (np. embedded C++)

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++
Funkcje uogólnione - wzorce

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