



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Zaawansowane programowanie w języku C++ Wyjątki

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



Politechnika Łódzka

Politechnika Łódzka, ul. Żeromskiego 116, 90-924 Łódź, tel. (042) 631 28 83
www.kapitalludzki.p.lodz.pl



Sytuacje wyjątkowe

- Wyjątkiem jest nieoczekiwana sytuacja, której nie da się sensownie obsłużyć w danym miejscu w kodzie:
 - brak pamięci, wyjście poza indeks kontenera itd.
- Tradycyjne podejście to przekazanie przez funkcję informacji o błędzie jako wartości zwracanej:
 - Informacja o błędzie tego samego typu co wartość zwracana
 - A co kiedy funkcja wykorzystuje cały zakres typu zwracanego?
 - Konieczność sprawdzenia wartości zwracanej po każdym wywołaniu funkcji



Przykład

```
#include <new>
```

```
MyClass *mc = new( std::nothrow ) MyClass;
```

```
if (!mc)
{
    std::cout << "Brak pamięci!" << std::endl;
    exit(1);
}
```



Wyjątki w C++

- W C++ wyjątki się „rzuca” (ang. throw)
- Rzucony wyjątek musi być „przechwycony” (ang. catch)
- Jeżeli używamy jakiejś funkcji mogącej generować wyjątki to w C++ mówimy, że „próbujemy” ich użyć (ang. try)
- W C++ wyjątkiem jest obiekt

```
try {  
    kod, który podowuje wyjątek  
} catch ( wyjątek1 ) {  
    obsługa wyjątku1  
} catch ( wyjątek2 ) {  
    obsługa wyjątku2  
} catch ( ... ) {  
    obsługa pozostałych wyjątków  
}
```





Przykład bez przechwytywania wyjątków

```
#include<vector>

int main()
{
    std::vector<int> vec( 10, 1 );
    vec.at( 100 );
    return 0;
}
```





Prawidłowe przechwycenie wyjątku

```
#include<vector>
#include<stdexcept>
#include<iostream>

int main()
{
    std::vector<int> vec( 10, 1 );

    try {
        vec.at( 100 );
    } catch ( std::out_of_range& e ) {
        std::cout << "Zły indeks!\n";
        std::cout << e.what() << std::endl;
    }
    return 0;
}
```





Generowanie wyjątku

- Wyjątek może być obiektem dowolnej klasy
- Najczęściej wyjątek rzucany jest jako obiekt tymczasowy

```
throw exception();  
throw MyClass("Mój wyjątek");
```





Propagacja wyjątku

- Wyjątek zwija stos tak długo aż napotka na blok, który go obsłuży (przechwyci)
- Sprzątanie po wyjątku realizuje się poprzez destruktory!
- Jeżeli wyjątek nie zostanie przechwycony to program zakończy swoje działanie!





Ograniczenie generowanych wyjątków

- Z projektowego punktu widzenia korzystne jest jawne zdefiniowanie wyjątków jakie dana funkcja może rzucić:

```
void myFunc() throw ( std::out_of_range, myException );
```

- Funkcja z ograniczoną listą wyjątków może generować jedynie te wyjątki, które pojawiły się na liście
- Jeżeli rzucony przez funkcję zostanie wyjątek, którego nie ma na liście to wywołana będzie funkcja unexpected(), która domyślnie woła funkcję terminate()





Wyjątki a konstruktory i destruktory ...





KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Zaawansowane programowanie w języku C++ Wyjątki

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



Politechnika Łódzka

Politechnika Łódzka, ul. Żeromskiego 116, 90-924 Łódź, tel. (042) 631 28 83
www.kapitalludzki.p.lodz.pl