

---

# Programowanie Obiektowe w C++

---

## LISTA 3 (40 PKT)

### DEADLINE'Y DLA GRUP: OSTATNIE ZAJĘCIA W SEMESTRZE

#### ZADANIA:

1. (10 pkt) Za pomocą typu wyliczeniowego zdefiniuj dowolne cztery typy błędów. Pomocne może okazać się zapoznanie się z klasami wyjątków biblioteki standardowej (STL) dziedziczącymi klasę `std::exception`.

Zaprojektuj i zaimplementuj abstrakcyjną klasę WYJATEK reprezentującą nadrzędną klasę do obsługi wyjątków. Klasa, oprócz podstawowych składowych, musi zawierać 2 czysto wirtualne metody:

- a) do wyświetlania komunikatu błędu,
  - b) do określania typu błędu (na podstawie wspomnianego wyżej typu wyliczeniowego).
2. (15 pkt) Zaprojektuj i zaimplementuj trzy klasy pochodne w stosunku do klasy Wyjatek, reprezentujące trzy dowolne błędy. Każdy z błędów powinien posiadać:
    - a) informację na temat swojego typu (korzystając ze zdefiniowanego typu wyliczeniowego),
    - b) komunikat wypisywany podczas przechwycenia wyjątku,
    - c) przynajmniej jeden dodatkowy parametr, określany dla wyjątku podczas jego wywoływania.

Przykładowo, dla klasy reprezentującej wyjątek nadpisania tablicy poza jej zakresem:

- a) typem może być `out_of_range`,
- b) komunikatem - *Próba zapisu do tablicy poza zakresem.*,
- c) dodatkowym parametrem przechowywanym w klasie - indeks elementu, pod którym próbowano coś zapisać,
- d) innym parameterem - wielkość tablicy.

Dla klas, zapewnij konstruktory uzupełniające dodatkowe parametry.

3. (15 pkt) Zaprezentuj działanie zaimplementowanych klas wyjątków na dowolnym przykładzie, z wykorzystaniem bloków `try{} catch{}`. Dodatkowe parametry klas rzucanych wyjątków uzupełniaj w konstruktorach, w momencie ich wywoływania w funkcji `throw()`. Przechwytuj wyjątki w odpowiednio zdefiniowanych funkcjach `catch()`. Dla każdego przechwyconego wyjątku, wykorzystaj jego specyficzne parametry, typ i tekst komunikatu do wyświetlenia na konsoli błędów komunikatu zbiorczego na temat całego zdarzenia. Sprawdź, czy wyjątki możesz także wywoływać bądź przechwytywać za pomocą klasy bazowej WYJATEK.

4. DODATKOWE - na dodatkowe punkty lub wyższą ocenę (zadania 1-3 muszą być wykonane!)

Do realizacji podobnego jak wyżej zadania i budowy klas wyjątków wykorzystaj bibliotekę `Boost Exception` i klasę `boost::exception`. Informacji o aktywnych, przechwyconych błędach dostarczaj z wykorzystaniem następujących metod:

- `boost::error_info`
- `boost::exception/operator«`
- `boost::tuple/operator«`
- `boost::get_error_info`
- `boost::enable_error_info`

Do generowania wyjątków powinieneś wykorzystać metody:

- `BOOST_THROW_EXCEPTION`
- `throw_exception`

Boost udostępnia także pewne zdefiniowane typy informacji o błędach:

- `errinfo_api_function`
- `errinfo_at_line`
- `errinfo_errno`
- `errinfo_file_handle`
- `errinfo_file_name`
- `errinfo_file_open_mode`
- `errinfo_nested_exception`
- `errinfo_type_info_name`