

Przyjąć, że udostępniona jest przestrzeń nazw std, std::placeholders i boost**Zadanie 1 (5pkt)**

W systemie uruchamia się wiele translacji, każda w oddzielnym wątku. Obserwujemy słabą skalowalność. Proszę usunąć problem. Przykład użycia poniżej.

```
struct Document {
    boost::mutex m_;
    std::string str_;
};
typedef std::shared_ptr<Document> PDoc;
struct Dict {
    boost::mutex m_;
    map<string, string> dict_;
    string translate(const string& word) {
        auto it = dict_.find(word);
        if(it != dict_.end())
            return (*it).second;
        else
            return word;
    }
};
typedef std::shared_ptr<Dict> PDict;

void zad1() {
    PDict dictPlEn(new Dict);
    PDict dictPlDe(new Dict);
    PDoc docA(new Document);
    PDoc docB(new Document);
    //pomijam inicjacje slownikow
    //pomijam inicjacje dokumentow

    ThreadTranslate t1(docA, dictPlEn);
    ThreadTranslate t2(docA, dictPlDe);
    ThreadTranslate t3(docB, dictPlEn);
    ThreadTranslate t4(docB, dictPlDe);

    try {
        boost::thread thrd1( ref(t1) );
        boost::thread thrd2( ref(t2) );
        boost::thread thrd3( ref(t3) );
        boost::thread thrd4( ref(t4) );
        thrd1.join(); thrd2.join();
        thrd3.join(); thrd4.join();
    } catch (exception&) {
        cout << "error" << endl;
    }
}
```

```
class ThreadTranslate {
public:
    ThreadTranslate(PDoc doc, PDict dict)
        : doc_(doc), dict_(dict) {}

    void operator() () {
        boost::mutex::scoped_lock lock_doc(doc_>m_);
        boost::mutex::scoped_lock lock_dict(dict_>m_);

        vector<string> tokens;
        string token;
        stringstream is(doc_>str_);
        while( is >> token) {
            tokens.push_back(token);
        }

        transform( tokens.begin(),
                   tokens.end(),
                   tokens.begin(),
                   [&](string word){return dict_>translate(word); } );

        std::ostringstream os;
        copy( tokens.begin(),
              tokens.end(),
              ostream_iterator<string>(os, "_"));
        out = os.str();
    }

    string getTranslation() { return out; }
private:
    PDoc doc_;
    PDict dict_;
    string out;
};
```

Zadanie 2 (5pkt)

Zaproponuj szablon median, który zwraca liczbę środkową ze swoich argumentów. Przykład zastosowania: median(1, 3, 2) zwraca wartość 2.

```
template<typename T> T median(T a, T b, T c) {
    BOOST_CONCEPT_ASSERT(( boost::LessThanComparableConcept<T>));
```

Notatki / uwagi do prowadzącego

Zadanie 3 (3pkt)

NAME to Twoje nazwisko zapisane wielkimi literami ASCII (zamiast 'Ą' jest 'A'), np. WROZKA dla Wróżka.

```
const string NAME = "_____";
```

Podaj napis, który zostanie wydrukowany przez funkcję zad3

```
typedef boost::variant<int, string> V;
int count(const std::vector<V>& v) {
    class Vis : public boost::static_visitor<void> {
    public:
        Vis() : max_(0) {}
        void operator()(const int& i){ if(i > max_) max_ = i; }
        void operator()(const std::string& s) {
            try {
                max_ = boost::lexical_cast<int>(s);
            } catch(boost::bad_lexical_cast &) { }
        }
        int max_;
    } vis;
    for_each(v.begin(), v.end(), [&](const V& v){apply_visitor(vis, v)});
    return vis.max_;
}
void zad3() {
    std::vector<V> v;
    v.push_back(NAME.size()); v.push_back(1); v.push_back("5"); v.push_back(NAME);
    cout << count(v) << flush;
}
```

Zadanie 4 (4pkt)

Podaj napis, który zostanie wydrukowany przez funkcję zad4. Stałą NAME zdefiniowano w poprzednim zadaniu.

```
template <unsigned n> int sum(int k) { return n + sum<n-1>(k); }
template <> int sum<1>(int k) { return k; }
template <> int sum<0>(int k) { return 0; }
void zad4() {
    cout << sum<2>(NAME.size()) << ";" << sum<10>(NAME.size());
}
```

Zadanie 5 (5pkt)

Dostarcz funkcję count_in_range, która zwraca ilość liczb z wektora, które mają wartość z danego zakresu (większe równe min i mniejsze równe max). Jeżeli kolekcja jest pusta zwracamy 0. Użyj algorytmów z biblioteki standardowej.

```
int most_common(const std::vector<int>& numbers, int min, int max) {
```

Pytanie 1 (1pkt)

Co dają testy jednostkowe?

Pytanie 2 (1pkt)

Zaproponuj zagadnienie, które Twoim zdaniem warto byłoby omówić na przedmiocie ZPR

Pytanie 3 (1pkt)

Ile godzin w semestrze poświęciłeś na przedmiot ZPR (wykład, projekt, kolokwia, nauka własna i inne)

Ile godzin w semestrze poświęciłeś na realizację projektu z ZPR