

Proszę wpisywać odpowiedzi w miejscach na to przeznaczonych

Zadanie 1 (2pkt)

Dana jest funkcja:

```
std::unique_ptr<Event> Service::method(double d, const Other* obj) {  
    if ((obj == nullptr) || !obj->isEnabled())  
        return std::make_unique<IgnoredValueEvent>(d);  
    if (d < obj->minimumSupportedValue())  
        throw InvalidArgument(d);  
    return (obj->threshold() < d) ? std::make_unique<LimitReachedEvent>(d) : nullptr;  
}
```

Ile przypadków testowych należy napisać, by uzyskać:

- 100% pokrycia linii kodu,
- 100% pokrycia gałęzi kodu (mierzonych narzędziem w rodzaju LCOV).

Podaj wartości argumentów konieczne do uzyskania wspomnianych pokryć.

Zadanie 2 (2pkt)

Proszę odpowiedzieć na poniższe pytania:

- Co pokrycie kodu mówi o jakości testu?

- Czy (i ewentualnie jak) można mierzyć "utrzymywalność" (maintability) kodu?

Zadanie 3 (2pkt)

Co jest nie tak z poniższym kodem?

```
int Client::download(const Item& i, Processing p, bool confirmed)
{
    // sprawdzamy łąpoczenie
    if (isConnected() && isActive() && (requests.size() == 0) && ((security == nullptr) || security->isAdmin() || security->allowsDownloads()))
    {
        auto tmp = i.options.flags[3].value;
        if (tmp < 15) {
            tmp = 2 * i.size();
            switch (p) {
                case Processing::Async: return confirmed ? startDownload(i, tmp) : - 5;
                case Processing::Sync: return confirmed ? downloadNow(i) : -3;
                default: return 0;
            }
        } else {
            switch (p) {
                case Processing::Sync: return confirmed ? downloadNow(i) : -3;
                case Processing::Async: return confirmed ? startDownload(i, tmp - 2) : - 5;
                default: return 1;
            }
        }
    }
    else
    {
        if (parent.mainProvider.registry.items.cache.contains(i))
            return 2;
        return -4;
    }
    return -5;
}
```

Proszę opisać jak należałoby poprawić ten kod (jakie operacje i w jaki sposób wykonane, kod nie jest konieczny).

Uwagi do prowadzącego (K. Grochowski):

Proszę wpisywać odpowiedzi w miejscach na to przeznaczonych

Zadanie 4 - współbieżne wzorce projektowe (2pkt)

Proszę podać zawartość zmiennej counter po wykonaniu kodu pokazanego w kolumnie 1. Jeżeli nie może wystąpić wyścig, to wartość minimalna = wartość maksymalna.

kod	wartość minimalna	wartość maksymalna
counter = 0; async_no_mutex();		
counter = 0; async_mutex();		
counter = 0; thread_no_mutex();		
counter = 0; thread_mutex();		

```
int counter = 0;
mutex m;

void serve( int i) { counter += i; }
void serve_m( int i) {
    lock_guard l(m);
    serve(i);
}
void serve_t(int i) {
    this_thread::sleep_for(chrono::milliseconds(10));
    serve(i);
}
void serve_tm(int i) {
    this_thread::sleep_for(chrono::milliseconds(10));
    serve_m(i);
}

void async_no_mutex() {
    boost::asio::io_service io;
    boost::asio::steady_timer t1(io, chrono::milliseconds(10) );
    boost::asio::steady_timer t2(io, chrono::milliseconds(10) );
    boost::asio::steady_timer t3(io, chrono::milliseconds(10) );
    t1.async_wait([&](const boost::system::error_code& error){ serve(1);});
    t2.async_wait([&](const boost::system::error_code& error){ serve(2);});
    t3.async_wait([&](const boost::system::error_code& error){ serve(3);});
    io.run();
}

void async_mutex() {
    boost::asio::io_service io;
    boost::asio::steady_timer t1(io, chrono::milliseconds(10) );
    boost::asio::steady_timer t2(io, chrono::milliseconds(10) );
    boost::asio::steady_timer t3(io, chrono::milliseconds(10) );
    t1.async_wait([&](const boost::system::error_code& error){ serve_m(1);});
    t2.async_wait([&](const boost::system::error_code& error){ serve_m(2);});
    t3.async_wait([&](const boost::system::error_code& error){ serve_m(3);});
    io.run();
}

void threads_no_mutex() {
    thread th1( serve_t, 1 ); thread th2( serve_t, 2 ); thread th3( serve_t, 3 );
    th1.join(); th2.join(); th3.join();
}

void threads_mutex() {
    thread th1( serve_tm, 1 ); thread th2( serve_tm, 2 ); thread th3( serve_tm, 3 );
    th1.join(); th2.join(); th3.join();
}
```

Uwagi do prowadzącego (R. Nowaka):

Zadanie 5 - współbieżne wzorce projektowe (2pkt)

Popraw aplikację pokazaną poniżej. Kodu poza ramką proszę nie poprawiać, nie jest istotny.

```
using Data = int; //symuluje dane do obrbki
static const Data END_DATA = 0;
static const Data VALID_DATA = 1;
int main() {
    DataSynch data;
    data.data_ = 1;
    Writer w1(data);
    Reader r1(data), r2(data), r3(data);
    thread th1( ref(w1) );
    thread th2( ref(r1) ), th3( ref(r2) ), th4( ref(r3) );
    th1.join(); th2.join(); th3.join(); th4.join();
    return 0;
}
```

```
struct DataSynch { //dane do obrbki w roznnych watkach
    Data data_;
    mutex read_;
    mutex write_;
};

struct Reader {
    Reader(DataSynch& data) : data_(data) {}

    void operator()() {
        bool finish = false;
        while(!finish) {
            lock_guard<mutex> lock(data_.read_);
            //tylko czyta wspoldzielone dane
            finish = (data_.data_ == END_DATA);
        }
    }
    DataSynch& data_;
};

struct Writer {
    Writer(DataSynch& data) : data_(data) {}

    void operator()() {
        static const int NUM = 10; //symuluje wielkosc danych do obrbki
        for(int i=0;i<NUM;++i) {
            lock_guard<mutex> lock2(data_.write_);
            //pisze wspoldzielone dane
            data_.data_ = VALID_DATA; //symuluje modyfikacje danych
        }
        //koniec zapisu, wstawiana dana 'pusta'
        {
            lock_guard<mutex> lock2(data_.write_);
            data_.data_ = END_DATA;
        }
    }
    DataSynch& data_;
};
```