

Proszę wpisywać odpowiedzi w miejscach na to przeznaczonych

### Zadanie 1 (2pkt)

Proszę odpowiedzieć na poniższe pytania:

- Jak ocenić jakość testu? Na ile ocena będzie obiektywna?

10. The following table summarizes the results of the study. The first column lists the variables, the second column lists the sample size, and the third column lists the estimated effect sizes.

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

## Zadanie 2 (2pkt)

Dana jest funkcja:

```
    std::optional<Value> Service::method(const Other* obj, std::string_view name) {
        if ((name.size() == 0 )|| (obj == nullptr))
            throw InvalidArgument();
        if (obj->isEmpty())
            return this->forceDefaults() ? std::optional<Value>{} : std::nullopt;
        return obj->property(name);
    }
```

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 3 (2pkt)

Co jest nie tak z poniższym kodem?

```
int Client::upload(const Item& i, Processing p, bool async)
{
    // sprawdzamy rozpoczęcie
    if (isConnected() && isAuthorized() && (itemsQueue.size() == 0)
        && ((security == nullptr) || security->allowsUploads() || security->isAdmin()))
    {
        auto tmp = i.options[2].flags.value;

        if (tmp < 15) {
            tmp = 2 * i.size();
            switch (p) {
                case Processing::Async: return async ? startUpload(i, tmp) : -5;
                case Processing::Sync: return async ? -3 : uploadNow(i);
                default: return 0;
            }
        } else {
            switch (p) {
                case Processing::Async: return async ? startUpload(i, tmp - 2) : -5;
                case Processing::Sync: return async ? -3 : uploadNow(i);
                default: return 0;
            }
        }
    }
    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. Grochowskiego):

**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;<br>thread_no_mutex(); |                   |                    |
| counter = 0;<br>thread_mutex();    |                   |                    |
| counter = 0;<br>async_no_mutex();  |                   |                    |
| counter = 0;<br>async_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 threads_no_mutex() {
    thread th1( serve_t, 1 ); thread th2( serve_t, 2 );
    th1.join(); th2.join();
}
void threads_mutex() {
    thread th1( serve_tm, 1 ); thread th2( serve_tm, 2 );
    th1.join(); th2.join();
}
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) );
    t1.async_wait([&](const boost::system::error_code& error){ serve(1);});
    t2.async_wait([&](const boost::system::error_code& error){ serve(2);});
    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) );
    t1.async_wait([&](const boost::system::error_code& error){ serve_m(1);});
    t2.async_wait([&](const boost::system::error_code& error){ serve_m(2);});
    io.run();
}

```

---

**Uwagi do prowadzącego (R. Nowaka):**

## Zadanie 5 - współbieżne wzorce projektowe (2pkt)

Jakie zjawisko występuje w poniższym kodzie (wyścig/zakleszczenie/zagłodzenie/żadno z powyższych)?

Popraw kod w ramce. Kodu poza ramką proszę nie poprawiać, nie jest istotny.

```
using Data = int;
const Data EMPTY = -1; //reprezentuje brak danych
int main () {
    Process p1(100), p2(200), p3(300);
    p1.setPrev(&p3);
    p2.setPrev(&p1);
    p3.setPrev(&p2);
    thread thrd1( ref(p1) );
    thread thrd2( ref(p2) );
    thread thrd3( ref(p3) );
    thrd1.join();
    thrd2.join();
    thrd3.join();
    return 0;
}
```

```
class Process {
public:
    static const int NUM = 100; //liczba operacji

    Process(int id) : id_(id) {
        data_.push_back(id_);
        data_.push_back(id_);
    }
    void setPrev(Process* p) { prev_ = p; }

    Data get() {
        Data ret = EMPTY;
        lock_guard<mutex> lock(m_);
        if(!data_.empty() ) {
            ret = data_.front();
            data_.pop_front();
        }
        return ret;
    }
    void operator()() {
        if(!prev_) return; //nie ma zrodla danych

        for(int i=0;i<NUM;++i) {
            lock_guard<mutex> lock(m_);
            Data d = prev_->get();
            if( d != EMPTY ) {
                data_.push_back(d+1);

            }
        }
    }
private:
    int id_;
    Process* prev_ = nullptr;
    list<Data> data_;
    mutex m_;
};
```