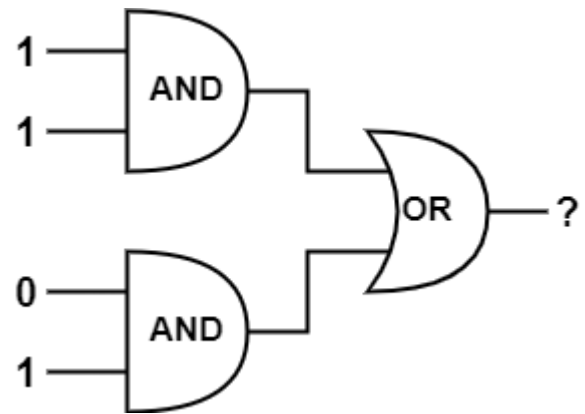


Пример испита из предмета "Објектно оријентисано програмирање 1"

1. Систем за симулацију рада дигиталног логичког кола је састављеног од логичких елемената као што су AND, OR, NOT, XOR, као и од улазних сигнала и излазних чворова. Сваки логички елемент има један или више улаза и тачно један излаз. Сваки чвор може бити повезан са другим чворовима, тако да излаз једног елемента постаје улаз другог. Треба обезбедити једноставан начин за конфигурацију кола додавањем елемената у коло и повезивањем улаза и излаза, постављањем вредности на улазне сигнале и израчунавањем вредности на излази. Типови логичких елемената и начини израчунавања њихових излаза су различити и могу се мењати и проширивати у будућности. Распоредити описане одговорности по класама и скицирати све класе потребне за описани механизам, са потпуном имплементацијом овог механизма. Показати употребу овог механизма на примеру кола са слике и исписати вредност која се појављује на излазу.



2. Прилагодити следећи програмски сегмент принципима објектно оријентисаног програмирања. Претпоставити да су све функције које се користе претходно декларисане.

...

```
bool status = validateCircuit(circuit);
if (status == true) {
    status = printOutput(circuit);
    if (status == true) {
        // do something
    } else {
        std::cout << "Output error" << std::endl;
    }
} else {
    std::cout << "Configuration error" << std::endl;
}
...
```

3. Навести грешку која постоји у следећем програму (навести да ли је грешка у превођењу, повезивању или извршавању и дати њено објашњење), а потом исправити ову грешку.

```
// math.h
class Math {
public:
    static int square(int x);
};

// main.cpp
#include "math.h"
#include <iostream>

int main() {
    std::cout << Math::square(5) << std::endl;
    return 0;
}
```

4. Која варијанта функције `A::getNumber()` се позива у главном програму уколико је показивач `a` дефинисан као:

а) `A* a = new A(1);`

б) `const A* a = new A(1);`

в) `A* const a = new A(1);`

Код:

```
class A {
public:
    A(int n) : number(n) {}
    const int getNumber() const { return number; }
    int& getNumber() { return number; }
private:
    int number;
};

int main() {
    // definicija pokazivača „a“
    a->getNumber();
    return 0;
}
```

5. Дат је део дефиниције класе `Log` која представља систем за логовање порука. Имплементирати операцију `write` класе `Log` која додаје текстуалну поруку у интерну листу тако да ту операцију могу позивати само операције класе `DebugTool` и класе изведене из класе `Log`.

```
class Log {  
    ...  
private:  
    std::vector<std::string> messages;  
};
```

6. Класу `x` опремити тако да се њени објекти могу поредити на једнакост и неједнакост (`==`, `!=`). Објекти класе `x` су једнаки ако су једнаке вредности које се налазе на придруженим адресама.

```
class X {  
    ...  
private:  
    int* address;  
};
```