

## 1 Specificați și testați funcția: (1.5p)

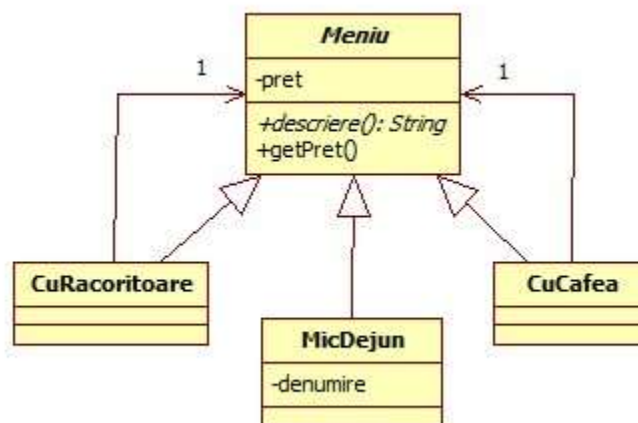
```
std::pair<int, int> f(std::vector<int> l) {  
    if (l.size() < 2) throw std::exception{};  
    std::pair<int, int> rez{-1, -1};  
    for (auto el: l) {  
        if (el < rez.second) continue;  
        if (rez.first < el) {  
            rez.second = rez.first;  
            rez.first = el;  
        } else {  
            rez.second = el;  
        }  
    }  
    return rez;  
}
```

2 Indicați rezultatul execuției pentru următoarele programe c++. Dacă sunt erori indicați locul unde apare eroarea și motivul.

```
//2 a (1p)  
#include <iostream>  
#include <vector>  
struct A {  
    A() {std::cout << "A";}   
    virtual void print() {  
        std::cout << "A";  
    }  
};  
struct B : public A {  
    B() { std::cout << "B"; }  
    void print() override {  
        std::cout << "B";  
    }  
};  
int main() {  
    std::vector<A> v;  
    v.push_back(A{});  
    v.push_back(B{});  
    for (auto& el : v) el.print();  
    return 0;  
}
```

```
//2 b (0.5p)  
#include <iostream>  
using namespace std;  
class A {  
    int x;  
public:  
    A(int x) : x{ x } {}  
    void print(){cout<< x <<endl;}  
};  
A f(A a) {  
    a.print();  
    a = A{ 10 };  
    a.print();  
    return a;  
}  
int main() {  
    A a{ 4 };  
    a.print();  
    f(a);  
    a.print();  
}
```

3 Scrieți codul C++ ce corespunde diagramei de clase UML. (4p)



- Clasa abstracta **Menu** are o metoda pur virtuala descriere()
- **CuRacoritoare** si **CuCafea** conțin un meniu si metoda descriere() adaugă textul “cu racoritoare” respectiv “cu cafea” la descrierea meniului conținut. Prețul unui meniu care conține răcoritoare creste cu 4 Ron, cel cu cafea costa in plus 5 RON.
- Clasa **MicDejun** reprezintă un meniu fără răcoritoare si fără cafea, metoda descriere() returnează denumirea meniului. In restaurant pizzeria exista 2 feluri de mic dejun: Ochiuri si Omleta, la prețul de 10 respectiv 15 RON.

Se cere:

- 1 Codul C++ **doar pentru clasele: Menu, CuCafea (0.75)**
  - 2 Scrieți o funcție C++ care returnează o lista de meniuri: un meniu cu Omleta cu răcoritoare si cafea, un meniu cu Ochiuri si cafea, un meniu cu Omleta. **(0.5p)**
  - 3 In programul principal se creează o comanda (folosind funcția descrisa mai sus), apoi se tipărește descrierea si prețul pentru fiecare pizza in ordinea descrescătoare a preturilor. **(0.25p)**
- Creați doar metode si attribute care rezultă din diagrama UML (adăugați doar lucruri specifice C++ ex: constructori). Nu adăugați câmpuri, metode, nu schimbați vizibilitatea, nu folosiți friend. Folosiți STL unde există posibilitatea.

Detalii barem: **1.5p** Polimorfism, **1p** Gestiunea memoriei, **1.5p** Restul(defalcăt mai sus)

4 Definiți clasa Measurement astfel încât următoarea secvență C++ sa fie corecta sintactic si sa efectueze ceea ce indica comentariile. (2p)

```
int main() {
    //creaza un vector de masuratori cu valorile (10,2,3)
    std::vector<Measurement<int>> v{ 10,2,3 };
    v[2] + 3 + 2; //aduna la masuratoarea 3 valoarea 5
    std::sort(v.begin(), v.end()); //sorteaza masuratorile
    //tipareste masuratorile (in acest caz: 2,8,10)
    for (const auto& m : v) std::cout << m.value() << ", ";
    return 0;
}
```