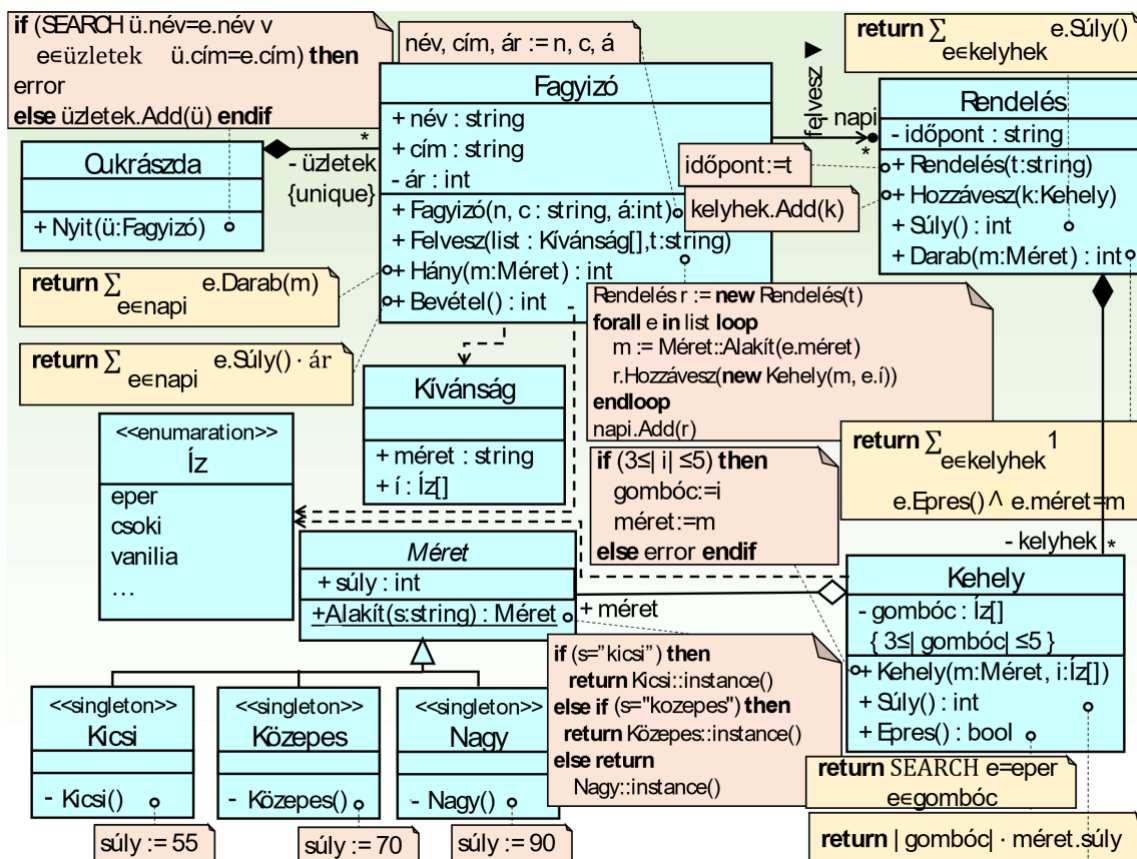


Modellezzük egy cukrász cég fagyizóinak működését. A fagyizók nevét és címét ismerjük, nevük egyedi. A fagyizókban háromféle fagyalt-kehely rendelhető: kicsi, közepes és nagy. A fagyalt-kehelyek háromöt gombócból állnak, a gombóc súlya kicsi esetén 55g, közepes esetén 70g, a nagy pedig 90g, a gombócok mind más ízűek, de a megrendelő választhatja ki az ízeiket. A fagyizóban nyilvántartják az aznapi kínálatot (milyen ízű fagyiból lehet választani), valamint az aznapi rendeléseket. A cég fagyizói egymástól eltérő áron adhatják a fagyit, amelynek ára annak ízétől független, és egy napon belül nem változik. Egy megrendelést az időpontja azonosít.

- Hány epret is tartalmazó fagyalt-kehelyet adtak el az elsőként megadott fagyizóban közepes méretű kehelyből?
- Mennyi az elsőként megadott fagyizó napi bevétele?



Közepes szint:

Implementálja a feladat mellékelt modelljét a sárga színű metódustörzsek nélkül, majd a saját kódját a megadott main.cpp fájlal szerkessze össze, és próbálja ki.

Miután feltöltötte a munkáját a TMS-be és úgy látja, hogy a programja működik, szóljon az egyik felügyelő tanárnak, hogy ellenőrizze a programját.

c:\Kleon_egyetem\000tantargyak\2_felev\OEP_programozos\masodik_mintaZH\fagy\cukraszda.cpp

```
#include "cukraszda.h"

void Cukraszda::Nyit(Fagyizo* u){
    for(Fagyizo* e : uzletek)
    {
        if(e->get_cim()==u->get_cim())
        {
            throw exception();
        }
    }
    uzletek.push_back(u);
}
```

c:\Kleon_egyetem\000tantargyak\2_felev\OEP_programozos\masodik_mintaZH\fagy\cukraszda.h

```
#pragma once
#include "rendeles.h"
#include "kivansag.h"
#include "meret.h"
#include "iz.h"
#include "fagyizo.h"
#include "kehely.h"

#include <vector>
#include <string>

class Meret;
class Fagyizo;
class Kivansag;
class Kehely;
class Rendeles;

using namespace std;

class Cukraszda{
private:
    vector<Fagyizo*> uzletek;
public:
    void Nyit(Fagyizo* u);
    vector<Fagyizo*> get_uzletek(){return uzletek;}

};
```

c:\Kleon_egyetem\000tantargyak\2_felev\OEP_programozos\masodik_mintaZH\fagy\fagyizo.cpp

```
#include "fagyizo.h"

void Fagyizo::Felvesz(vector<Kivansag> /*&*/list,string t){
    Rendeles* r = new Rendeles(t);
    for(Kivansag e : list)
    {
        Meret* m = Meret::Atalakit(e.get_meret());
        r->Hozzavesz(new Kehely(m,e.get_iz()));
    }
    napi.push_back(r);
}
```

c:\Kleon_egyetem\000tantargyak\2_felev\OEP_programozos\masodik_mintaZH\fagyiz\fagyizo.h

```
#pragma once
#include "rendeles.h"
#include "kivansag.h"
#include "meret.h"
#include "iz.h"
#include "kehely.h"
#include "cukraszda.h"

#include <vector>
#include <string>

class Meret;
class Kivansag;
class Cukraszda;
class Kehely;
class Rendeles;

using namespace std;

class Fagyizo{
private:
    string nev,cim;
    int ar;
    vector<Rendeles*> napi;
public:
    Fagyizo(string n, string c,int ar):nev(n),cim(c),ar(ar){}
    ~Fagyizo(){}

    void Felvesz(vector<Kivansag> /*&*/list,string t);

    vector<Rendeles*> get_napi(){return napi;}
    string get_cim(){ return cim;}
    string get_nev(){return nev;}
    int get_ar(){return ar;}

};
```

c:\Kleon_egyetem\000tantargyak\2_felev\OEP_programozos\masodik_mintaZH\fagyiz\iz.h

```
#pragma once
#include "rendeles.h"
#include "kivansag.h"
#include "meret.h"
#include "kehely.h"
#include "fagyizo.h"
#include "cukraszda.h"

#include <vector>
#include <string>

class Meret;
class Fagyizo;
class Cukraszda;
class Kivansag;
class Kehely;
class Rendeles;

using namespace std;

typedef enum Iz{
    eper,csoki,vanilia
} Iz;
```

c:\Kleon_egyetem\000tantargyak\2_felev\OEP_programozos\masodik_mintaZH\fagyiz\Kehely.cpp

```
#include "kehely.h"

Kehely::Kehely(Meret* m, vector<Iz> i){
    if(i.size()<3 ||i.size()>5)
    {
        throw exception();
    }
    gomboc=i;
    meret=m;
}
```

c:\Kleon_egyetem\000tantargyak\2_felev\OEP_programozos\masodik_mintaZH\fagy\kehely.h

```
#pragma once
#include "rendeles.h"
#include "kivansag.h"
#include "meret.h"
#include "iz.h"
#include "fagyizo.h"
#include "cukraszda.h"

#include <vector>
#include <string>
class Fagyizo;
class Cukraszda;
class Kivansag;
class Meret;
class Rendeles;

using namespace std;

class Kehely{
private:
    vector<Iz> gomboc;
    Meret* meret;

public:
    Kehely(Meret* m, vector<Iz> i);
    Meret* get_meret()
    {
        return meret;
    }
};
```

c:\Kleon_egyetem\000tantargyak\2_felev\OEP_programozos\masodik_mintaZH\fagy\kivansag.h

```
#pragma once
#include "rendeles.h"
#include "kehely.h"
#include "meret.h"
#include "iz.h"
#include "fagyizo.h"
#include "cukraszda.h"

#include <vector>
#include <string>

class Meret;
class Fagyizo;
class Cukraszda;
class Kehely;
class Rendeles;

using namespace std;

class Kivansag{
public:
    string meret;
    vector<Iz> i;
    Kivansag()=default;
    ~Kivansag(){}

    vector<Iz> get_iz(){return i;}
    string get_meret(){return meret;}
};
```

c:\Kleon_egyetem\000tantargyak\2_felev\OEP_programozos\masodik_mintaZH\fagy\main.cpp

```
#include "cukraszda.h"
#include <iostream>
#include <string>

#include <cassert>

int main()
{
    Cukraszda* c = new Cukraszda();
    c->Nyt(new Fagyizo("Fagyizo", "Varos-Utca-Hazszam",100));

    Fagyizo* f = c->get_uzletek()[0];

    std::cout << f->get_nev() << std::endl;
    std::cout << f->get_cim() << std::endl;
    std::cout << f->get_ar() << std::endl;

    Kivansag ki1, ki2, ki3;
    ki1.i = {Iz::csoki,Iz::csoki,Iz::vanilia};
    ki1.meret = "kicsi";

    ki2.i = {Iz::csoki,Iz::eper,Iz::vanilia,Iz::eper};
    ki2.meret = "nagy";

    ki3.i = {Iz::csoki,Iz::csoki,Iz::vanilia,Iz::vanilia,Iz::eper};
    ki3.meret = "kozepes";

    std::vector<Kivansag> kList1;
    kList1.push_back(ki1);
    kList1.push_back(ki2);
    kList1.push_back(ki3);

    f->Felvesz(kList1,"10:00");
    std::cout << f->get_napi()[0]->get_idopont() << std::endl;

    Rendezes* r1 = f->get_napi()[0];

    Kehely* ke1 = r1->get_kelyhek()[0];
    Kehely* ke2 = r1->get_kelyhek()[1];
    Kehely* ke3 = r1->get_kelyhek()[2];

    assert(ke1->get_meret()->get_suly() == 55);
    assert(ke2->get_meret()->get_suly() == 90);
    assert(ke3->get_meret()->get_suly() == 70);

    std::cout << ke1->get_meret()->get_suly() << " - " << (ke1->get_meret() == Kicsi::instance()) << std::endl;
    std::cout << ke2->get_meret()->get_suly() << " - " << (ke2->get_meret() == Nagy::instance()) << std::endl;
    std::cout << ke3->get_meret()->get_suly() << " - " << (ke3->get_meret() == Kozepes::instance()) << std::endl;

    Kicsi::destroy();
    Kozepes::destroy();
    Nagy::destroy();
    return 0;
}
```

c:\Kleon_egyetem\000tantargyak\2_felev\OEP_programozos\masodik_mintaZH\fagy\meret.cpp

```
#include "meret.h"
Kicsi* Kicsi::_instance = nullptr;
Nagy* Nagy::_instance = nullptr;
Kozepes* Kozepes::_instance=nullptr;

Meret* Meret::Atalakit(string s)
{
    if(s=="kicsi")
    {
        return Kicsi::instance();
    }
    else if(s=="kozepes")
    {
        return Kozepes::instance();
    }
    else{
        return Nagy::instance();
    }
}
```

c:\Kleon_egyetem\000tantargyak\2_felev\OEP_programozos\masodik_mintaZH\fagy\meret.h

```
#pragma once

#include "rendeles.h"
#include "kivansag.h"
#include "kehely.h"
#include "iz.h"
#include "fagyizo.h"
#include "cukraszda.h"

#include <vector>
#include <string>

class Fagyizo;
class Cukraszda;
class Kivansag;
class Kehely;
class Rendeles;

using namespace std;

class Meret{
protected:
    int suly;
public:
    static Meret* Atalakit(string s);
    virtual int get_suly(){return suly;}
    virtual ~Meret(){}
};

class Kicsi: public Meret{
private:
    Kicsi()
    {
        suly=55;
    }
    static Kicsi* _instance;
public:
    static Kicsi* instance(){if(_instance==nullptr){_instance=new Kicsi();}return _instance;}
    static void destroy(){if(_instance!=nullptr){delete _instance;}_instance=nullptr;}
};

class Kozepes : public Meret{
private:
    Kozepes()
    {
        suly=70;
    }
    static Kozepes* _instance;
public:
    static Kozepes* instance(){if(_instance==nullptr){_instance=new Kozepes();}return _instance;}
    static void destroy(){if(_instance!=nullptr){delete _instance;}_instance=nullptr;}
};

class Nagy : public Meret{
private:
    Nagy(){
        suly=90;
    }
    static Nagy* _instance;
public:
    static Nagy* instance(){if(_instance==nullptr){_instance=new Nagy();}return _instance;}
    static void destroy(){if(_instance!=nullptr){delete _instance;}_instance=nullptr;}
};
```

c:\Kleon_egyetem\000tantargyak\2_felev\OEP_programozos\masodik_mintaZH\fagy\rendeles.h

```
#pragma once
#include "kehely.h"
#include "kivansag.h"
#include "meret.h"
#include "iz.h"
#include "fagyizo.h"
#include "cukraszda.h"

#include <vector>
#include <string>

class Meret;
class Fagyizo;
class Kivansag;
class Kehely;
class Cukraszda;

using namespace std;

class Rendeles{
private:
    vector<Kehely*> kelyhek;
    string idopont;
public:
    Rendeles(string t):idopont(t){}
    void Hozzavesz(Kehely* k){
        kelyhek.push_back(k);
    }
    vector<Kehely*> get_kelyhek(){return kelyhek;}
    string get_idopont()
    {
        return idopont;
    }
};
```