

Фирмата **FINKI Cloud** има потреба од решение за управување со image датотеките од различните оперативни системи кои ги користат при сетирање на нови сервери. За таа цел потребно е да се дефинираат следниве класи:

OperativenSistem

Во класата **OperativenSistem** треба да се чуваат следниве податоци:

- име на оперативниот систем (динамички алоцирана низа од знаци)
- верзија (float)
- тип (енумерација со можни вредности **LINUX**, **UNIX**, **WINDOWS**)
- големина (во GB) (float)

Дополнително, во рамките на класата потребно е да се дефинира:

- конструктори со и без параметри
- copy constructor
- destructor
- преоптоварување на операторот **=**
- метод за печатење на информациите - **void pecati()** (видете го излезот од тест примерите за структурата)
- метод за проверка на **еднаквост** помеѓу два оперативни системи (два оперативни системи се еднакви ако имаат исто име, верзија, тип и големина) - **bool ednakviSe(const OperativenSistem &os)**
- метод за споредба на **верзии** помеѓу два оперативни система - **int sporediVerzija(const OperativenSistem &os)**, кој ќе враќа 0 доколку верзиите се исти, -1 доколку верзијата на оперативниот систем од аргументот е поголема и 1 во преостанатиот случај.
- метод за проверка дали два оперативни системи се од иста **фамилија**. Два оперативни системи се од иста фамилија доколку имаат исто име и се од ист тип. **bool istaFamilija(const OperativenSistem &sporedba)**

Repozitorium

Во класата **Repozitorium** треба да се чуваат следниве податоци:

- име на репозиториумот (низа од 20 знака)
- динамички алоцирана низа од `OperativenSistem`
- број на оперативни системи кои моментално се наоѓаат во репозиториумот (int)

Дополнително, во рамките на класата потребно е да се дефинира:

- конструктор `Repozitorium(const char *ime)`
- деструктор
- метод `void pecatiOperativniSistemi()` кој ќе ги печати сите оперативни системи кои се дел од репозиториумот
- метод `void izbrishi(const OperativenSistem &operativenSistem)` кој ќе го избрише оперативниот систем проследен како аргумент, доколку тој се наоѓа во репозиториумот. Да се користи методот `ednakviSe` за споредба на два оперативни система.
- метод за додавање на оперативен систем (`void dodadi(const OperativenSistem &nov)`) кој ќе го има следново однесување:
 - Доколку веќе постои оперативен систем од иста фамилија (да се користи методот за проверка на фамилија) кој е постар од проследениот оперативен систем, тогаш постоечкиот оперативен систем се заменува со новиот.
 - Во секој друг случај, проследениот оперативен систем се додава како дополнителен во репозиториумот.

For example:

Input	Result
-------	--------

FINKICloud	Repozitorium: FINKICloud
d	Ime: Ubuntu Verzija: 17.04 Tip: 1
2	Golemina:2.25GB
Ubuntu	Ime: Ubuntu Verzija: 16.04 Tip: 1
17.04	Golemina:1.25GB
1	====Brishenje na operativen sistem====
2.25	Repozitorium: FINKICloud
Ubuntu	Ime: Ubuntu Verzija: 17.04 Tip: 1
16.04	Golemina:2.25GB
1	Ime: Ubuntu Verzija: 16.04 Tip: 1
1.25	Golemina:1.25GB
Ubuntu	
14.04	
1	
1.25	