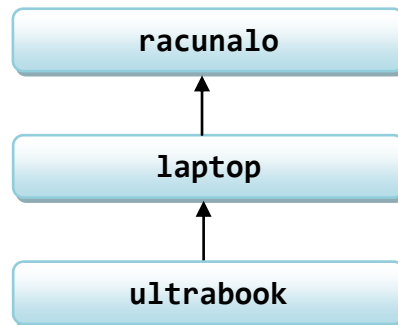


Zadatak:

Napišite sučelje i implementaciju za klase **racunalo**, **laptop** i **ultrabook**. Sljedeći dijagram opisuje relacije među klasama. Sučelja za klase spremite u datoteku **racunalo.h**, a implementacije u datoteku **racunalo.cpp**.



Klasa racunalo

Ova klasa opisuje računalo. Svako računalo ima tri parametra koji ga određuju: brzina procesora (MHz), količina rama (MB) i veličina ekrana (inč).

- **racunalo(int brzina, int ram, float velicinaEkрана)**
Konstruktor za klasu racunalo.
- **int brzina()**
Vraća brzinu procesora.
- **int ram()**
Vraća količinu memorije.
- **float velicinaEkрана()**
Vraća veličinu ekrana.
- **int cijenaNadogradnje(int novaBrzina, int noviRam);**
*Vraća cijenu nadogradnje računala ako ugradimo novi procesor s brzinom novaBrzina i novi ram s količinom noviRam. Cijenu računamo po formuli: $(novabrzina - starabrzina) * 2 + (noviram - stariram) * 1$*
- **racunalo& nadjiPrviVeciEkran();**
Vraća referencu na iduće računalo s većim ekranom (s obzirom na sva računala koja trenutno postoje i programu), tj. od svih računala koja imaju strogo veći ekran od računala koje je pozvalo funkciju vraća ono s najmanjim ekranom. Ukoliko postoji više rješenja vraća se proizvoljno rješenje. Ukoliko ne postoji takvo računalo vraća se referenca na samo sebe.
- **static racunalo& najbrzeRacunalo();**
Vraća referencu na najbrže računalo (s obzirom na sva vozila koja trenutno postoje u programu). Najbrže računalo je ono s najbržim procesorom. Ukoliko ima više takvih, vraća se proizvoljno rješenje.
- **static racunalo* spojiRacunala(racunalo &r1, racunalo &r2)**
Vraća pokazivač na novo kreirano računalo. Novo računalo se sastoji od komponenti računala r1 i r2. Uzima se brži procesor, veći ram i veći ekran. Iz računala iz kojeg se uzima komponenta, ta komponenta se postavlja na 0. Ako su neke vrijednosti iste u oba računala uzima se iz računala r1.

Klasa laptop

Ova klasa opisuje laptop. Laptop ima i parametar mase.

- **laptop(int brzina, int ram, float velicinaEkрана, float masa)**
Konstruktor za klasu laptop.

- **float masaLaptopa()**
Vraća masu laptopa.
- **int cijenaNadogradnje(int novaBrzina, int noviRam)**
*Vraća cijenu nadogradnje laptopa ako ugradimo novi procesor s brzinom novaBrzina i novi ram s količinom noviRam. Cijenu računamo po formuli: $(novabrzina - starabrzina) * 3 + (noviram - stariram) * 2$*

Klasa ultrabook

Ova klasa opisuje ultrabook. Ultrabook ima i parametar debljine.

- **ultrabook(int brzina, int ram, float velicinaEkрана, float masa, int debljina)**
Konstruktor za klasu ultrabook.
- **int debljina()**
Vraća debljinu ultrabooka.
- **int cijenaNadogradnje(int novaBrzina, int noviRam)**
*Vraća cijenu nadogradnje laptopa ako ugradimo novi procesor s brzinom novaBrzina i novi ram s količinom noviRam. Cijenu računamo po formuli: $(novabrzina - starabrzina) * 3 + (noviram - stariram) * 3$*

Primjer klijentskog programa

```
#include <iostream>
#include "racunalo.h"

int main(void)
{
    racunalo A(1800, 1024, 17), B(2500, 2048, 23), C(2200, 4096, 24);
    racunalo *nizRacunala[5];
    nizRacunala[0] = new laptop(2000, 2048, 15.6, 5.5);
    nizRacunala[1] = new racunalo(3300, 1024, 19);
    nizRacunala[2] = new ultrabook(1500, 1024, 15.6, 2.3, 22);
    nizRacunala[3] = new ultrabook(1700, 2048, 13.3, 1.8, 18);
    nizRacunala[4] = new laptop(2800, 4096, 17.3, 6);

    racunalo &najbrze = racunalo::najbrzeRacunalo();
    cout << najbrze.brzina() << " " << najbrze.ram() << " " << najbrze.velicinaEkрана() << endl;
    // 3300 1024 19

    racunalo &veciEkran = A.nadjiPrviVeciEkran();
    cout << veciEkran.brzina() << " " << veciEkran.ram() << " " << veciEkran.velicinaEkрана() << endl;
    // 2800 4096 17.3

    for(int i = 0; i < 5 ; i++)
        cout << nizRacunala[i]->cijenaNadogradnje(3500, 4096) << endl;
    // 8596
    // 3472
    // 15216
    // 11544
    // 2100

    ultrabook U(1500, 512, 13.3, 2.1, 20);
    cout << U.brzina() << " " << U.ram() << " " << U.velicinaEkрана() << " " << U.masaLaptopa() << " " <<
    U.debljina() << endl;
    // 1500 512 13.3 2 20

    racunalo X(1800, 1024, 23), Y(2200, 1024, 21);
    racunalo *Z = racunalo::spojiRacunala(X, Y);
    cout << X.brzina() << " " << X.ram() << " " << X.velicinaEkрана() << endl;
    cout << Y.brzina() << " " << Y.ram() << " " << Y.velicinaEkрана() << endl;
    cout << Z->brzina() << " " << Z->ram() << " " << Z->velicinaEkрана() << endl;
    // 1800 0 0
    // 0 1024 21
    // 2200 1024 23

    return 0;
}
```

Opće napomene

- Struktura, funkcije i datoteke koje šalžete moraju se zvati točno onako kako je zadano u zadatku. Pazite na mala i velika slova!
- Trebate poslati samo sučelje i implementaciju. U datotekama koje šalžete ne smije se nalaziti funkcija `main()`!
- nijedna funkcija ne smije ništa učitavati s tipkovnice ili neke datoteke, niti išta ispisivati na ekran ili u neku datoteku.
- Svaki od `main`-ova pomoću kojih testiramo ispravnost vašeg programa neće pozivati sve gore navedene funkcije. Stoga, ako neku od funkcija ne znate napisati ipak možete dobiti koji bod (u tom slučaju tu funkciju nemojte navesti niti u `.h` niti u `.cpp` datoteci ili napravite neku trivijalnu implementaciju).

Ispravnost implementacija koje napišete bit će provjerena tako da ćemo mi napisati razne klijentske programe koji će deklarirati nekoliko varijabli zadane strukture, i na njima pozivati funkcije koje ste trebali napisati. Ako se poslani programi ne budu uspješno povezivali (linkali) s našim klijentskim programima, smatrat će se neispravnima.

Neki klijentski programi provjeravat će samo neke jednostavnije funkcije, dok će neki provjeravati sve funkcije koje trebate napisati. Provjera je potpuno automatska, tako da je od presudne važnosti da se pridržavate specifikacije. Nepridržavanje lako može uzrokovati osvojenih 0 bodova iz zadaće!

Naravno, za provjeru radi li implementacija prije nego što je pošaljete, preporučuje se da je testirate pomoću nekog klijentskog programa. No taj klijentski program ne šalžete!

Za sva pitanja vezana uz ovu zadaću javite se asistentu Vujčiću na ivujcic+rp1@gmail.com.