Programowanie Obiektowe

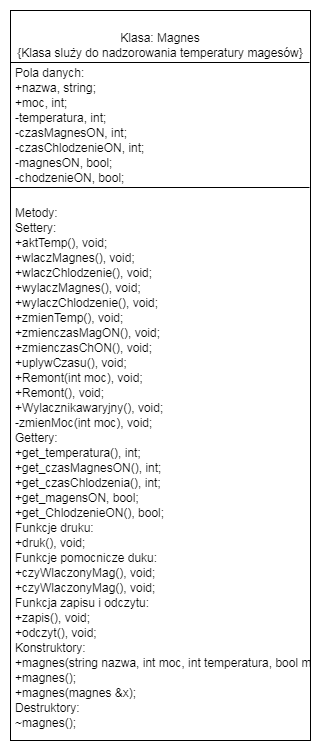
Tytuł: Magnes

Wykonał: Jakub Gałuszka

Grupa: 41M2

Semestr:2019/2020

1. **Diagram UML (1 klasa)**

****

1. **Kod programu**

#include <iostream>

#include <fstream>

using namespace std;

class magnes {

public:

**//pola z danymi publicznymi**

string nazwa;

int moc;

**//settery**

void aktTemp(){

temperatura=temperatura-czasChlodzenieON\*2+czasMagnesON;

if (temperatura<1){

temperatura=1;

}

}

void wlaczMagnes(){

magnesON=1;

cout<<"Wlaczono magnes."<<endl;

}

void wylaczMagnes(){

magnesON=0;

czasMagnesON=0;

cout<<"Wylaczono magnes."<<endl;

}

void wlaczChlodzenie(){

chlodzenieON=1;

cout<<"Wlaczono chlodzenie."<<endl;

}

void wylaczChlodzenie(){

chlodzenieON=0;

czasChlodzenieON=0;

cout<<"Wylaczono chlodzenie."<<endl;

}

void zmienTemp(int temperatura){

this->temperatura=temperatura;

if (temperatura<1){

temperatura=1;

}

}

void zmienczasMagON(int czasMagnesON){

this->czasMagnesON=czasMagnesON;

cout<<"Zmieniono czas uruchomienia magnesu."<<endl;

}

void zmienczasChON(int czasChlodzenieON){

this->czasChlodzenieON=czasChlodzenieON;

cout<<"Zmieniono czas trwania chlodzenia."<<endl;

}

void uplywCzasu(int czas){

if (magnesON==1){

czasMagnesON=czasMagnesON+czas;

}

if (chlodzenieON==1){

czasChlodzenieON=czasChlodzenieON+czas;

}

}

void Remont(int moc){

cout << "Wykonano remont. Zmieniono moc magnesu. Na czas remontu wylaczono magnes i chlodzenie" << endl;

zmienMoc(moc);

wylaczMagnes();

wylaczChlodzenie();

}

**//polimorfizm**

void Remont(){

cout << "Wykonano drobny remont magnesu. Na czas remontu wylaczono magnes i chlodzenie" << endl;

wylaczMagnes();

wylaczChlodzenie();

}

**//gettery**

int get\_temperatura(){

return temperatura;

}

int get\_czasMagnesON(){

return czasMagnesON;

}

int get\_czasChlodzenia(){

return czasChlodzenieON;

}

bool get\_magnesON(){

return magnesON;

}

bool get\_chlodzenieON(){

return chlodzenieON;

}

**//funkcje pomocnicze do druku**

void czyWlaczonyMag(){

if (this->magnesON==0){

cout<<"wylaczony";

}

else {

cout<<"wlaczony";

}

}

void czyWlaczonyCh(){

if (this->chlodzenieON==0){

cout<<"wylaczone";

}

else {

cout<<"wlaczone";

}

}

**//funkcja druku**

void druk(){

aktTemp();

cout <<"Nazwa: " << nazwa <<" Moc: " << moc << " Temperatura: " << get\_temperatura() << " Stan magnesu: "; czyWlaczonyMag(); cout << " od " << get\_czasMagnesON() << " sekund "<<endl;

cout << "Stan chlodzenia: "; czyWlaczonyCh(); cout<< " od " << get\_czasChlodzenia() << " sekund " << endl;

czasMagnesON=0; **//zerowanie czasu magnesu**

czasChlodzenieON=0; **//zerowanie czasu chłodzenia**

}

**//zapis do pliku**

void zapis(){

ofstream plik("plik.txt", ios\_base::app);

plik << nazwa << " "<< moc << " " << get\_temperatura() << " " << get\_magnesON() << " " << get\_chlodzenieON() << " " << get\_czasMagnesON() << " " << get\_czasChlodzenia()<<endl;

plik.close();

cout<<"ZAPISANO: "<< nazwa<<endl;

}

**//odczyt z pliku**

void odczyt(){

ifstream plik;

plik.open ("plik.txt");

plik >> nazwa >> moc >> temperatura >> magnesON >> chlodzenieON >> czasMagnesON >> czasChlodzenieON;

cout<<"ODCZYT: "<<nazwa<<endl;

cout <<"Nazwa: " <<nazwa <<" Moc: " << moc << " Temperatura: "<< temperatura << " Stan magnesu: "; czyWlaczonyMag();

cout << " od " << czasMagnesON << " sekund "<<endl << "Stan chlodzenia: "; czyWlaczonyCh();

cout << " od " <<czasChlodzenieON << " sekund " << endl << "KONIEC OCZYTU: "<<nazwa <<endl;

plik.close();

}

**//metoda wylacznik awaryjny**

void Wylacznikawaryjny(){

cout << "Awaryjnie wylaczono magnes i chlodzenie." << endl;

wylaczMagnes();

wylaczChlodzenie();

}

**//konstruktory\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**//konstruktor 1**

magnes(string nazwa, int moc, int temperatura, bool magnesON, int chlodzenieON){

this->nazwa=nazwa;

this->moc=moc;

this->temperatura=temperatura;

this->magnesON=magnesON;

this->chlodzenieON=chlodzenieON;

czasMagnesON=0;

czasChlodzenieON=0;

cout << "Utworzono magnes."<<endl;

}

**//konstruktor 2**

magnes(){

this->nazwa="domyslna";

this->moc=1;

this->temperatura=293;

this->magnesON=0;

this->chlodzenieON=0;

czasMagnesON=0;

czasChlodzenieON=0;

cout << "Utworzono domyslny magnes."<<endl;

}

**//konstruktor kopiujacy**

magnes(magnes &x) {

nazwa=x.nazwa;

moc=x.moc;

temperatura=x.temperatura;

magnesON=x.magnesON;

chlodzenieON=x.chlodzenieON;

czasMagnesON=x.czasMagnesON;

czasChlodzenieON=x.czasChlodzenieON;

cout << "Utworzono drugi taki sam magnes."<<endl;

}

**//destruktor**

~magnes(){

cout << "Zdemontowano magnes: "<<endl;

druk();

}

protected:

//- private

//+ public

//# protected

private:

**//pola z danymi prywatnymi**

int temperatura, czasMagnesON, czasChlodzenieON;//chlodzenie (wlacz wylacz, wylacznik bezpieczenstwa,

bool magnesON, chlodzenieON;

**//metoda zmieniajaca moc**

void zmienMoc(int moc){

this->moc=moc;

}

};

int main()

{

magnes a1("LHC1",12,200,1,1), a2(a1);

magnes \*a3 = new magnes; **//dynamiczny przydzial pamieci**

a1.druk();

a1.uplywCzasu(10);

a1.druk();

a1.wylaczChlodzenie();

a1.druk();

a1.wlaczChlodzenie();

a1.druk();

a1.zapis();

a1.odczyt();

delete a3; **//zwolnienie pamieci**

return 0;

};