Programowanie Obiektowe

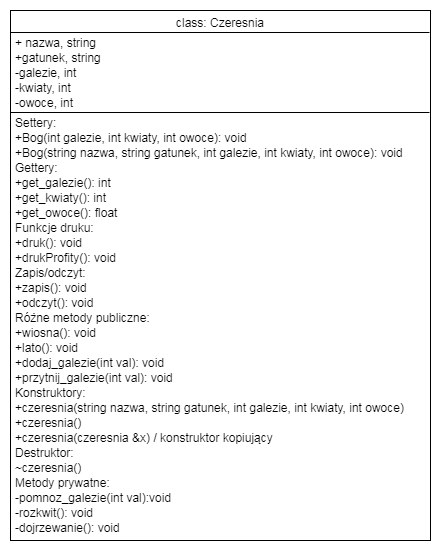
Projekt 1

Tytuł: Czereśnia

Wykonał: Jakub Gałuszka

Grupa: 41M2

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

****

1. **Kod źródłowy programu**

#include <iostream>

#include <fstream>

using namespace std;

int deski=0, zbiory=0;

class czeresnia {

public:

**//pola z danymi publicznymi**

string nazwa, gatunek;

**//settery**

void Bog(int galezie, int kwiaty, int owoce){

cout << "Bog moze zmienic wszystkie atrybuty czeresni." << endl;

this->galezie=galezie;

this->kwiaty=kwiaty;

this->owoce=owoce;

}

**//polimorfizm**

void Bog(string nazwa, string gatunek, int galezie, int kwiaty, int owoce){

cout << "Bog moze zmienic wszystkie atrybuty czeresni." << endl;

this->galezie=galezie;

this->kwiaty=kwiaty;

this->owoce=owoce;

this->nazwa=nazwa;

this->gatunek=gatunek;

}

**//gettery**

int get\_galezie(){

return galezie;

}

int get\_kwiaty(){

return kwiaty;

}

float get\_owoce(){

return owoce;

}

**//funkcje druku**

void druk(){

cout<<"Nazwa: " << nazwa <<" Gatunek: " << gatunek << " Galezie: " << get\_galezie() << " Kwiaty: " << get\_kwiaty() << " Owoce: " << get\_owoce() << endl;

}

void drukProfity(){

cout << "Deski: " << deski << " Zbiory: " << zbiory << endl;

}

**//zapis do pliku**

void zapis(){

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

plik << nazwa << " "<< gatunek << " " << get\_galezie() << " " << get\_kwiaty() << " " << get\_owoce() << endl;

plik.close();

}

**//odczyt z pliku**

void odczyt(){

ifstream plik;

plik.open ("plik.txt");

plik >> nazwa >> gatunek >> galezie >> kwiaty >> owoce;

cout <<"Nazwa: " <<nazwa <<" Gatunek: " << gatunek << " Galezie: "<< galezie << " Kwiaty: " << kwiaty << " Owoce: " <<owoce << " TO JEST ODCZYT" <<endl;

plik.close();

}

**//inne ciekawe metody**

**//metoda 1**

void wiosna(){

cout << "Przyszla wiosna, galezie puszczaja nowe pedy i zakwitaja. Na wiosne na galeziach nie ma owocow." << endl;

pomnoz\_galezie(2);

rozkwit();

this->owoce=0;

**//metoda 2**

}

void lato(){

cout << "Przyszlo lato, kwiaty przeradzaja sie w owoce." << endl;

dojrzewanie();

this->kwiaty=0;

}

**//metoda 3**

void dodaj\_galezie(int val){

this->galezie=get\_galezie()+val;

cout << "Dodano galezie." << endl;

}

**//metoda 4**

void przytnij\_galezie(int val){

this->galezie=get\_galezie()-val;

cout << "Przycieto galezie." << endl;

}

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

**//konstruktor 1**

czeresnia(string nazwa, string gatunek, int galezie, int kwiaty, int owoce){

this->galezie=galezie;

this->kwiaty=kwiaty;

this->owoce=owoce;

this->nazwa=nazwa;

this->gatunek=gatunek;

cout << "Posadzono czeresnie."<<endl;

}

**//konstruktor 2**

czeresnia(){

this->galezie=1;

this->kwiaty=0;

this->owoce=0;

this->nazwa="domyslna";

this->gatunek="domyslny";

cout << "Posadzono czeresnie o jednej galezi (pniu)."<<endl;

}

**//konstruktor kopiujacy**

czeresnia(czeresnia &x) {

galezie=x.galezie;

kwiaty=x.kwiaty;

owoce=x.owoce;

nazwa=x.nazwa;

gatunek=x.gatunek;

cout << "Posadzono druga taka sama czeresnie."<<endl;

}

**//destruktor**

~czeresnia(){

deski=deski+get\_galezie();

zbiory=zbiory+get\_owoce();

cout << "Scieto czeresnie "<<nazwa<<", z kazdej galezi jest jedna deska. Zebrano owoce."<< endl;

drukProfity();

}

protected:

private:

**//pola z danymi prywatnymi**

int galezie, kwiaty, owoce;

**//metoda 5 (prywatna)**

void pomnoz\_galezie(int val){

this->galezie=get\_galezie()\*val;

}

**//metoda 6 (prywatna)**

void rozkwit(){

this->kwiaty=get\_galezie();

}

**//metoda 7 (prywatna)**

void dojrzewanie(){

this->owoce=get\_kwiaty();

}

};

int main()

{

czeresnia a1("Czeresnia1","Burlat",1,2,3), a3(a1);

czeresnia \*a4 = new czeresnia(); **//dynamiczny przydzial pamieci**

a1.druk();

a1.dodaj\_galezie(3);

a1.druk();

a1.przytnij\_galezie(1);

a1.druk();

a1.wiosna();

a1.druk();

a1.lato();

a1.druk();

a1.Bog(1,1,1);

a1.druk();

delete a4; **//zwolnienie pamieci**

czeresnia a2(a1);

a2.druk();

a1.zapis();

a1.odczyt();

return 0;

}