1.

#include <iostream>

using namespace std;

template<class T>

class Kolekcija

{

public:

T\* niz;

int max;

int trenutno;

Kolekcija(int max = 10) {

}

~Kolekcija() {

}

bool ukloni() {

//uklanja zadnji element iz niza

}

bool ukloni(T el) {

//Uklanja element koji je jednak proslijeđenom

}

bool ukloni(int pozicija) {

//uklanja element na određenoj poziciji

}

void dodaj(T el) {

}

bool prazan() {

//vrši provjeru da li je niz prazan

}

T\* GetNajmanjiElement() {

//vraća najmanji element iz niza i to kao pokazivač

}

//preklopiti operator <<

};

class Zivotinja

{

private:

char \*vrsta;

int tezina;

public:

Zivotinja() {

}

Zivotinja(char \*\_vrsta, int \_tezina) {

}

~Zivotinja() { }

//konstruktor kopije

//operator < koji poredi dvije životinje po težini

//operator =

//operator ==

//operator<< koji ispisuje vrstu i težinu životinje

//operator>> koji učitava vrstu i težinu životinje

};

void main() {

Zivotinja z1, z2, z3, z4;

cin >> z1 >> z2 >> z3>>z4;

Kolekcija<Zivotinja> zivotinje(2);

zivotinje.dodaj(z1);

zivotinje.dodaj(z2);

zivotinje.dodaj(z3);

zivotinje.dodaj(z4);

cout << zivotinje << endl<<endl;

Zivotinja \*min = zivotinje.GetNajmanjiElement();

if (min != nullptr) {

cout << "Životinja sa najmanjom težinom je: ";

cout << \*min << endl;

}

zivotinje.ukloni();

cout << "Nakon poziva funkcije ukloni(): \n";

cout << zivotinje << endl << endl;

zivotinje.ukloni(1);

cout << "Nakon poziva funkcije ukloni(1): \n";

cout << zivotinje << endl << endl;

zivotinje.ukloni(z1);

cout << "Nakon poziva funkcije ukloni(z1): \n";

cout << zivotinje << endl << endl;

delete min; min = nullptr;

system("pause>0");

}