**ZA1**

**triedy.h**

#include <iostream>

using namespace std;

class Zoo

{

private:

char miesto\_zoo[30];

char zviera[30];

int cis\_zvierata;

long int hodn\_z;

public:

Zoo() {}

Zoo(char \*miesto\_zoo, char \*zviera, int cis\_zvierata, int hodn\_z);

char \*VratMiesto\_zoo();

char \*VratZviera();

int VratCislo();

long int VratHodn\_z();

void ZmenMiesto\_zoo(char \*);

void ZmenZviera(char \*);

void ZmenCislo(int);

void ZmenHodn\_z(long int);

friend ostream& operator<<(ostream &vyst\_prud, Zoo z);

};

**triedy.cpp**

#include "triedy.h"

Zoo::Zoo(char \*m\_z, char \*z, int c\_z, int h\_z)

{

strcpy(miesto\_zoo, m\_z);

strcpy(zviera, z);

cis\_zvierata = c\_z;

hodn\_z = h\_z;

}

void Zoo::ZmenMiesto\_zoo(char \*mz)

{

strcpy(miesto\_zoo, mz);

}

void Zoo::ZmenZviera(char \*zv)

{

strcpy(zviera, zv);

}

void Zoo::ZmenCislo(int cis)

{

cis\_zvierata = cis;

}

void Zoo::ZmenHodn\_z(long int hz)

{

hodn\_z = hz;

}

char\* Zoo::VratMiesto\_zoo()

{

char \*pom;

pom = new char[20];

strcpy(pom, miesto\_zoo);

return pom;

}

char\* Zoo::VratZviera()

{

char \*pom;

pom = new char[20];

strcpy(pom, zviera);

return pom;

}

int Zoo::VratCislo()

{

return cis\_zvierata;

}

long int Zoo::VratHodn\_z()

{

return hodn\_z;

}

ostream& operator<<(ostream &vyst\_prud, Zoo z)

{

vyst\_prud << "miesto zoo: " << z.VratMiesto\_zoo() << endl

<< "zviera: " << z.VratZviera() << endl

<< "cislo zvierata: " << z.VratCislo() << endl

<< "spoloc. hodnota zvierata [EUR]: " << z.VratHodn\_z() << endl;

return vyst\_prud;

}

**hlavny\_zdroj\_sub.cpp**

#include "triedy.h"

int main()

{

char miesto[30], zviera[30];

int cislo;

long int hodnota;

int pocet;

int i = 0;

long int suma = 0;

Zoo objekty[10];

cout << "vlozte pocet vkladanych zvierat (max. 3): ";

cin >> pocet;

if ((pocet ==0) || (pocet > 3))

cout << "Zadali ste pocet zvierat mimo povoleneho rozsahu!" << endl;

else

{

while (i < pocet)

{

cout << endl << "vlozte miesto ZOO, kde zviera zije: ";

cin >> miesto;

cout << "vlozte druh zvierata: ";

cin >> zviera;

cout << "vlozte cislo zvierata: ";

cin >> cislo;

cout << "vlozte spolocensku hodnotu zvierata [EUR]: ";

cin >> hodnota;

objekty[i].ZmenMiesto\_zoo(miesto);

objekty[i].ZmenZviera(zviera);

objekty[i].ZmenCislo(cislo);

objekty[i].ZmenHodn\_z(hodnota);

i++;

}

cout << "----------------------------------------------" << endl;

cout << endl << "programom vytvoreny a inicializovany objekt zvierata pomocou parametrickeho konstruktora triedy 'Zoo': " << endl<<endl;

/\*Zoo Slon("Bojnice", "Slon", "1002", 11000);

cout << Slon; \*/

char miesto[] = "Bojnice"; char zviera[] = "slon";

Zoo Zviera(miesto, zviera, 1002, 11000);

cout << Zviera;

cout << endl<< "hodnoty instan. premennych objektov triedy 'Zoo' (zvierat), vlozene pouzivatelom programu: " << endl<<endl;

i = 0;

while (i < pocet)

{

cout << objekty[i] << endl;

suma += objekty[i].VratHodn\_z();

i++;

}

cout << endl;

suma += Zviera.VratHodn\_z();

cout << "celkova spolocenska hodnota uvedenych "<< i+1<<"-och zvierat [EUR]: " << suma << endl;

}

return 0;

}

ZA-1 inak

**triedy.h**

#include <iostream>

using namespace std;

class Zoo

{

private:

char Miesto[40];

char Druh[40];

int Cislo;

int Cena;

public:

Zoo() {}

Zoo(char \*Miesto, char \*Druh, int Cislo, int Cena);

char \*GetMiesto();

char \*GetDruh();

int GetCislo();

int GetCena();

void SetMiesto(char \*Place);

void SetDruh(char \*DruhZviera);

void SetCislo(int Number);

void SetCena(int Price);

};

**Triedy.cpp**

#include "triedy.h"

Zoo::Zoo(char \*Miesto, char \*Druh, int Cislo, int Cena)

{

strcpy(Zoo::Miesto, Miesto);

strcpy(Zoo::Druh, Druh);

Zoo::Cislo = Cislo;

Zoo::Cena = Cena;

}

void Zoo::SetMiesto(char \*Place)

{

strcpy(Miesto, Place);

}

char\* Zoo::GetMiesto()

{

char \*pom;

pom = new char[20];

strcpy(pom, Miesto);

return pom;

}

void Zoo::SetDruh(char \*DruhZviera)

{

strcpy(Druh, DruhZviera);

}

char\* Zoo::GetDruh()

{

char \*pom;

pom = new char[20];

strcpy(pom, Druh);

return pom;

}

void Zoo::SetCislo(int Number)

{

Cislo = Number;

}

int Zoo::GetCislo()

{

return Cislo;

}

void Zoo::SetCena(int Price)

{

Cena = Price;

}

int Zoo::GetCena()

{

return Cena;

}

**Source.cpp**

#include "triedy.h"

int main()

{

char Miesto[40], Druh[40];

int Cislo, Cena;

int Pocet;

int i = 0, j = 0;

Zoo objekty[10];

cout << "Vlozte pocet (max. 3): ";

cin >> Pocet;

if ((Pocet > 3) || (Pocet == 0))

cout << "Zly pocet" << endl;

else

{

while (i < Pocet)

{

i++;

cout << endl << "vlozte miesto zoo " << i << ". zvierata : ";

cin >> Miesto;

cout << "vlozte druh zvierata " << i << ". zvierata : ";

cin >> Druh;

cout << "vlozte cislo " << i << ". zvierata : ";

cin >> Cislo;

cout << "vlozte cenu " << i << ". zvierata : ";

cin >> Cena;

objekty[i].SetMiesto(Miesto);

objekty[i].SetDruh(Druh);

objekty[i].SetCislo(Cislo);

objekty[i].SetCena(Cena);

}

cout << endl << endl;

for (i = 1; i < Pocet + 1; i++)

{

cout << "Miesto : " << objekty[i].GetMiesto() << endl;

cout << "Druh : " << objekty[i].GetDruh() << endl;

cout << "Cislo : " << objekty[i].GetCislo() << endl;

cout << "Cena : " << objekty[i].GetCena() << endl;

cout << endl << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

}

}

return 0;

}

ZA2

#include<time.h>

#include<iostream>

using namespace std;

int najdi\_cisloB(int x, int pole[], int dlzka)

{

int najdene = 0;

int lavy = 0,pravy = dlzka-1,stred;

while ((lavy < pravy) && (najdene==0))

{

stred = (lavy + pravy) / 2;

if (pole[stred] == x)

najdene = 1;

else if (pole[stred] < x)

lavy = stred + 1;

else

pravy = stred - 1;

}

if (najdene == 1)

return stred;

else

return -1;

}

int main()

{

int i, n, m, index;

int cisla[101];

cout << "Vlozte velkost prehladavenj casti pola 'stat\_pole (max. 100)': ";

cin >> m;

cout << "\n";

srand((unsigned)time(NULL));

cisla[0] = (long)rand() % 80;

for (i = 1; i <= m; i++)

cisla[i] = cisla[i - 1] + rand() % 80;

for (i = 1; i <= m; i++)

cout << "stat\_pole [" << i - 1 << "] = " << cisla[i - 1] << endl;

cout << "\nvlozte hladane cislo v zobrazenej casti pola 'stat\_pole': ";

cin >> n;

index = najdi\_cisloB(n, cisla, m);

if (index == -1)

cout << "\nHladane cislo " << n << " sa v poli 'stat\_pole' NENACHADZA (zistovane binarnou funkciou)." << endl;

else

cout << "\nindex prvku pola 'stat\_pole' obsahujuci hladane cislo je (najdene binarnym vyhladavanim): "<<index<<endl;

return 0;

ZA 2

#include<time.h>

#include<iostream>

#define VYMEN(a, b) { long tmp = a; a = b; b = tmp; }

using namespace std;

/\*void QuickSort(long data[], int lavy, int pravy)

{

if (lavy < pravy)

{

int i = lavy, j = pravy, p = data[(lavy + pravy) / 2];

do

{

while (data[i] < p) i++;

while (p < data[j]) j--;

if (i <= j)

{

VYMEN(data[i], data[j]);

i++; j--;

}

} while (i <= j);

QuickSort(data, lavy, j);

QuickSort(data, i, pravy);

}

}

\*/

int NajdiCisloB(int velkost\_pola, int hladane\_cislo, long \*pole)

{

int lavy = 0, pravy = velkost\_pola - 1;

while (lavy <= pravy)

{

int stred = (lavy + pravy) / 2;

if (pole[stred] < hladane\_cislo)

{

lavy = stred + 1;

}

else if (pole[stred] > hladane\_cislo)

{

pravy = stred - 1;

}

else

{

return stred;

}

}

return -1;

}

int main()

{

long hladane\_cislo, velkost\_pola;

long stat\_pole[100];

int i = 0;

cout << "vlozte velkost prehladavanej casti pola \"stat\_pole\" max (100): ";

cin >> velkost\_pola;

srand((unsigned)time(NULL));

stat\_pole[0] = (long)rand() % 80;

for (i = 1; i < velkost\_pola; i++)

stat\_pole[i] = stat\_pole[i - 1] + rand() % 80;; //tato cast vygerenuje nahodne cisla do pola

//QuickSort(stat\_pole, 0, velkost\_pola - 1);

cout << velkost\_pola << " vygenerovanych prvkov pola \"stat\_pole\"" << endl;

for (i = 0; i < velkost\_pola; i++)

cout << "stat\_pole[" << i << "] = " << stat\_pole[i] << endl;

cout << endl;

cout << "vlozte hladane cislo v zobrazenej casti pola \"stat\_pole\":" << endl;

cin >> hladane\_cislo;

int index;

index = NajdiCisloB(velkost\_pola, hladane\_cislo, stat\_pole); // pocet prvkov, hladane cislo, tvoje stat\_pole

if (index > -1)

{

cout << "Index prvku pola \"stat\_pole\" obsahujuci hladane (najdene binarnym vyhladavanim) cislo: " << hladane\_cislo << " je : " << index << endl;

}

else

{

cout << "Hladane cislo: " << hladane\_cislo << " sa nenachadza v poli \"stat\_pole\"" << endl;

}

return 0;

}

**A2- ine**

#include<time.h>

#include<iostream>

using namespace std;

int vyhladavanie(int m, int n, long \*cisla)

{

int zaciatok = 1;

int koniec = m;

int pivot;

pivot = (zaciatok + koniec) / 2; // Find Mid Location of Array

while (zaciatok <= koniec && cisla[pivot] != n) // Compare Item and Value of Mid

{

if (cisla[pivot] < n)

zaciatok = pivot + 1;

else

koniec = pivot - 1;

pivot = (zaciatok + koniec) / 2;

}

if (cisla[pivot] == n)

{

cout << "\nNasla sa zhoda na indexe : " << pivot << endl;

}

else

{

cout << "Nenasla sa zhoda" << endl;

}

return pivot;

}

int main()

{

int i, n, m;

long cisla[101], \*hladane;

int zaciatok, koniec, pivot;

cout << "Vlozte velkost pola: ";

cin >> m;

srand((unsigned)time(NULL));

cisla[0] = (long)rand() % 80;

for (i = 1; i <= m; i++)

cisla[i] = cisla[i - 1] + rand() % 80;

for (i = 1; i <= m; i++)

cout << "vygenerovane cislo [" << i - 1 << "] : " << cisla[i - 1] << endl;

cout << "vlozte hladane cislo: ";

cin >> n;

int index;

index = vyhladavanie(m, n, cisla);

return 0;

}

**ZB1**

**triedy.h**

#include <iostream>

using namespace std;

class Automobil

{

private:

char miesto[50];

char znacka\_typ[50];

int rok;

int cislo;

long int cena;

public:

Automobil() {}

Automobil(char \*miesto, char \*znacka\_typ, int rok, int cislo,long int cena);

char \*VratMiesto();

char \*VratZnacku\_typ();

int VratRok();

int VratCislo();

long int VratCenu();

void ZmenMiesto(char \*);

void ZmenZnacka\_typ(char \*);

void ZmenRok(int);

void ZmenCislo(int);

void ZmenCenu(long int);

friend ostream& operator<<(ostream &vyst\_prud, Automobil a);

};

**triedy.cpp**

#include "triedy.h"

Automobil::Automobil(char \*m, char \*z\_t, int r, int c, long int ce)

{

strcpy(miesto, m);

strcpy(znacka\_typ, z\_t);

rok = r;

cislo = c;

cena = ce;

}

char \*Automobil::VratMiesto()

{

return miesto;

}

char \*Automobil::VratZnacku\_typ()

{

return znacka\_typ;

}

int Automobil::VratRok()

{

return rok;

}

int Automobil::VratCislo()

{

return cislo;

}

long int Automobil::VratCenu()

{

return cena;

}

void Automobil::ZmenMiesto(char \*mi)

{

strcpy(miesto, mi);

}

void Automobil::ZmenZnacka\_typ(char \*zn)

{

strcpy(znacka\_typ, zn);

}

void Automobil::ZmenRok(int r2)

{

rok = r2;

}

void Automobil::ZmenCislo(int cis)

{

cislo = cis;

}

void Automobil::ZmenCenu(long int cn)

{

cena = cn;

}

ostream& operator<<(ostream& vyst\_prud, Automobil a)

{

vyst\_prud << "miesto autobazaru s pred.autom :" << a.VratMiesto()

<< "znacka a typ auta:" << a.VratZnacku\_typ() << endl

<< "rok vyroby auta:" << a.VratRok() << endl

<< "predajne cislo auta:" << a.VratCislo() << endl

<< "predajna cena auta [EUR] :" << a.VratCenu()<< endl;

return vyst\_prud;

}

**hlavny\_zdroj\_sub.cpp**

#include "triedy.h"

int main()

{

char miesto[30], znacka[30];

int rok, cislo;

long int cena;

int pocet;

int i = 0;

long int suma = 0;

double priemer = 0;

Automobil objekty[10];

cout << "vlozte pocet vkladanych automobilov (max. 3): ";

cin >> pocet;

if ((pocet == 0) || (pocet > 3))

cout << "Zadali ste pocet automobilov mimo povoleneho rozsahu!" << endl;

else

{

while (i < pocet)

{

cout << endl << "vlozte miesto autobazara s predavanym automob: ";

cin >> miesto;

cout << "vlozte znacku a typ auta: ";

cin >> znacka;

cout << "vlozte rok vyroby auta: ";

cin >> rok;

cout << "vlozte predajne cislo auta: ";

cin >> cislo;

cout << "vlozte predajnu cenu auta [EUR]: ";

cin >> cena;

objekty[i].ZmenMiesto(miesto);

objekty[i].ZmenZnacka\_typ(znacka);

objekty[i].ZmenCislo(cislo);

objekty[i].ZmenRok(rok);

objekty[i].ZmenCenu(cena);

i++;

}

cout << "----------------------------------------------" << endl;

/\*Zoo Slon("Bojnice", "Slon", "1002", 11000);

cout << Slon; \*/

cout << endl << "programom vytvoreny a inicializovany objekt auta pomocou parametrickeho konstruktora triedy 'Automobil': " << endl << endl;

char miesto[] = "Trnava"; char znacka\_typ[] = "Nissan\_Primera";

Automobil Auto(miesto, znacka, 2006, 10020,3300);

cout << Auto;

cout << endl << "hodnoty instan. premennych objektov triedy 'Automobil' (aut), vlozene pouzivatelom programu: " << endl << endl;

i = 0;

while (i < pocet)

{

cout << objekty[i] << endl;

suma += objekty[i].VratCenu();

i++;

}

cout << endl;

suma += Auto.VratCenu();

priemer = (double)suma / (i + 1);

cout << "priemerna predajna cena uvedenych " << i + 1 << "-och aut [EUR]: " << priemer<< endl;

}

return 0;

}

**ZB2**

#include<time.h>

#include<iostream>

using namespace std;

int NajdiCisloL(int x, int pole[], int dlzka)

{

int i = 0;

while (i < dlzka && pole[i] != x)

i++;

if (i < dlzka)

return i;

else

return -1;

}

int main()

{

int i, n, m, index;

int cisla[101];

cout << "vlozte velkost prehladavanej casti pola 'stat\_pole (max. 100)': ";

cin >> m;

cout << "\n";

srand((unsigned)time(NULL));

for (i = 0; i <= m; i++)

{

cisla[i] = (long)rand() % 80;

cout << "stat\_pole [" << i << "] = " << cisla[i] << endl;

}

cout << "\nvlozte hladane cislo v zobrazenej casti pola 'stat\_pole': ";

cin >> n;

index = NajdiCisloL(n, cisla, m);

if (index == -1)

cout << "\nHladane cislo " << n << " sa v poli 'stat\_pole' NENACHADZA (zistene linearnou funkciou)." << endl;

else

cout << "\nindex prvku pola 'stat\_pole' obsahujuci hladane cislo je(najdene linearnym vyhladavanim): " << index << endl;

return 0;

}

B2

#include<time.h>

#define VYMEN(a, b) { long tmp = a; a = b; b = tmp; }

#include<iostream>

using namespace std;

int Vyhladavanie(int m, int n, long \*pole){

int i, s=0;

for (i = 0; i < m; i++)

{

if (pole[i] == n)

{

cout << "\nNasla sa zhoda na indexe : " << i+1 << endl;

s = 1;

break;

}

}

if (s == 0)

{

cout << "Nenasla sa zhoda" << endl;

return -1;

}

return i;

}

int main()

{

long n, s = 0, m = 20;

cout << "vlozte hladane cislo (1-100): ";

cin >> n;

long pole[20];

int i = 0;

srand((unsigned)time(NULL));

for (i = 0; i < 20; i++)

pole[i] = (double)rand() / (RAND\_MAX + 1) \* 100 + 1;

for (i = 0; i < 20; i++)

cout << "vygenerovane cisla [" << i + 1 << "] : " << pole[i] << endl;

cout << endl;

int index;

index = Vyhladavanie(20, n, pole); // pocet prvkov, hladane cislo, tvoje pole

cout << "i je : " << index;

return 0;

}

B2

#include<time.h>

#define VYMEN(a, b) { long tmp = a; a = b; b = tmp; }

#include<iostream>

using namespace std;

int NajdiCisloL(int velkost\_pola, int hladane\_cislo, long \*pole)

{

for (int i = 0; i < velkost\_pola; i++)

{

if (pole[i] == hladane\_cislo)

{

return i;

}

}

return -1;

}

int main()

{

long hladane\_cislo, velkost\_pola;

long stat\_pole[100];

int i = 0;

cout << "vlozte velkost prehladavanej casti pola \"stat\_pole\" max (100): ";

cin >> velkost\_pola;

srand((unsigned)time(NULL));

for (i = 0; i < velkost\_pola; i++)

stat\_pole[i] = (double)rand() / (RAND\_MAX + 1) \* 100 + 1; //tato cast vygerenuje nahodne cisla do pola

cout << velkost\_pola << " vygenerovanych prvkov pola \"stat\_pole\"" << endl;

for (i = 0; i < velkost\_pola; i++)

cout << "stat\_pole[" << i << "] = " << stat\_pole[i] << endl;

cout << endl;

cout << "vlozte hladane cislo v zobrazenej casti pola \"stat\_pole\":" << endl;

cin >> hladane\_cislo;

int index;

index = NajdiCisloL(velkost\_pola, hladane\_cislo, stat\_pole); // pocet prvkov, hladane cislo, tvoje stat\_pole

if (index > -1)

{

cout << "Index prvku pola \"stat\_pole\" obsahujuci hladane (najdene linearnym vyhladavanim) cislo: " << hladane\_cislo << " je : " << index << endl;

}

else

{

cout << "Hladane cislo: " << hladane\_cislo << " sa nenachadza v poli \"stat\_pole\"" << endl;

}

return 0;

}

C2

#include<iostream>

#include<time.h>

using namespace std;

int Naraznik(int x, int pole[], int dlzka)

{

int j = 0;

pole[dlzka] = x;

cout << pole[dlzka];

for (int i = 0; i < dlzka; i++)

{

if (pole[i] == x)

{

j = i;

}

}

if (j < dlzka)

{

return j;

}

}

int main()

{

srand((unsigned)time(NULL));

int n = 20;

int hladane;

int index;

int \*pole = new int[n];

cout << "Zadajte hladane cislo: -> ";

cin >> hladane;

for (int i = 0; i < n; i++)

{

pole[i] = rand() % 100;

}

cout << "\nZoznam prvkov pola" << endl << endl;

for (int i = 0; i < n; i++)

{

cout << "vygenerovane pole[" << i << "]: " << pole[i] << endl;

}

index = Naraznik(hladane, pole, n);

cout << pole[19] << endl;

cout << pole[20];

if (index > 0)

{

cout << "\nIndex cisla je: " << index << endl;

}

else

cout << "\nCislo sa v poli nenachadza!" << endl;

}