Zkouška z předmětu Jazyk C++ I 29. 5. 2012

Vytvořte systém řízení letového provozu s bázovou abstraktní třídou `Letadlo` a odvozenými třídami `CivilniLetadlo` a `VojenskeLetadlo`.

Třída `Letadlo` bude obsahovat:

- datovou složku `volaciKod` -- bude nastaveno v konstruktoru
- čistě virtuální metodu `PosliZpravu(const char* zprava)`
- parametrický konstruktor, který nastaví volací kód dle požadavku tvůrce objektu

Třída `CivilniLetadlo` bude navíc obsahovat počet pasažérů a zprávu bude posílat standardním komunikačním kanálem.

Třída `VojenskeLetadlo` bude zprávu posílat kódovaným komunikačním kanálem. Dále do této třídy přidejte výčet `typZbrani` s konstantami `Zadne`, `Konvencni`, `Jaderne`. A datovou složku tohoto typu.

Obě odvozené třídy budou mít pouze parametrické konstruktory, které inicializují (uživatelsky) kromě položek z bázové třídy taktéž počet pasažérů (pro `CivilniLetadlo`), nebo typ nesených zbraní (pro `VojenskeLetadlo`).

Komunikační kanál simulujte jen vytištěním zpráv do konzole. Dle vlastního uvážení názorně (viditelně u výstupu do konzole) rozlište kódovaný kanál. V hlavním programu vytvořte pole letadel a demonstrujte výhodu virtuálních metod zasláním společné zprávy všem letadlům.

Do tříd doplňte definici metody `ToString()`, která vrátí řetězec charakterizující letadlo. Funkčnost taktéž vyzkoušejte -- vytiskněte informace o všech letadlech na obrazovku. Zamyslete se nad tím, v jakých třídách bude nejvýhodnější implementaci provést.

Do tříd přetižte operátory porovnání (`==` a `!=`) a operátor součtu. Porovnání bude realizováno podle volacího kódu. Součet půjde provést jen u civilních letadel -- výsledkem součtu bude nová instance s volacím kódem prvního letadla a počtem pasažérů z obou letadel. Zamyslete se nad tím, do jakých tříd bude přetížení operátorů porovnání nejvýhodnější provést.

```
#include "stdafx.h"
#include <iostream>
#include <string>
#include <string.h>
#include <sstream>
using namespace std;

enum Velikost{SIZE = 6};

class Letadlo {

public:
	unsigned int volaciKod;
	virtual void PosliZpravu(const char* zprava) = 0;
	virtual string ToString() = 0;
	bool operator==(const Letadlo &druhy) const;
    bool operator!=(const Letadlo &druhy) const;
};

bool Letadlo::operator==(const Letadlo &druhy) const {
	if (volaciKod == druhy.volaciKod) {
	return true;}
	else {return false;}
}

bool Letadlo::operator!=(const Letadlo &druhy) const {
	if (volaciKod == druhy.volaciKod) {
	return false;}
	else {return true;}
}


class CivilniLetadlo : public Letadlo {
private:
	unsigned int pocetPasazeru;
public:
	void PosliZpravu(const char* zprava);
	CivilniLetadlo();
	CivilniLetadlo(unsigned int kod,unsigned int pocet) {volaciKod = kod; pocetPasazeru = pocet;}
	string ToString();
	CivilniLetadlo* operator+(const CivilniLetadlo &druhy);
};

CivilniLetadlo* CivilniLetadlo::operator+(const CivilniLetadlo &druhy) {
	CivilniLetadlo *civlet = new CivilniLetadlo(volaciKod, pocetPasazeru + druhy.pocetPasazeru);
	return civlet;
}

string CivilniLetadlo::ToString() {
	//cout << "Civilni letadlo\nKod: " << volaciKod << "\nPocet pasazeru: " << pocetPasazeru <<
"\n---------\n\n";
	stringstream ss;
	ss << "Civilni letadlo\nKod: " << volaciKod << "\nPocet pasazeru: " << pocetPasazeru <<
"\n---------\n\n";
	return ss.str();
}
```

```
void CivilniLetadlo::PosliZpravu(const char* zprava) {
	cout << "-bezna zprava-";
	cout << endl;
	cout << zprava;
	cout << endl;
	cout << "---konec bezne zpravy---\n";
	cout << endl;

}

class VojenskeLetadlo : public Letadlo {

public:
	enum typZbrani{ZADNE = 1, KONVENCNI = 2,JADERNE = 3};
	typZbrani typ;
	void PosliZpravu(const char* zprava);
	VojenskeLetadlo();
	VojenskeLetadlo(unsigned int kod,typZbrani typZ) {volaciKod = kod; typ = typZ;}
	string ToString();
};

void VojenskeLetadlo::PosliZpravu(const char* zprava) {
	cout << "############kodovany kanal############";
	cout << endl;
	cout << zprava;
	cout <<endl;
	cout << "##########################konec kodovane zpravy#########################\n";
	cout << endl;
}

string VojenskeLetadlo::ToString() {
	string str;
	switch(typ) {
	case 1:
		str = "ZADNE";
		break;
	case 2:
		str = "KONVENCNI";
		break;
	case 3:
		str = "JADERNE";
		break;
	}

	stringstream ss;
	ss << "Vojenske letadlo\nKod: " << volaciKod << "\nZbrane: " << str << "\n---------\n";
	return ss.str();

}

int _tmain(int argc, _TCHAR* argv[])
{

	Letadlo *pole[SIZE];
```

```
	CivilniLetadlo *cl1 = new CivilniLetadlo(1,100);
	CivilniLetadlo *cl2 = new CivilniLetadlo(2,200);
	CivilniLetadlo *cl3 = new CivilniLetadlo(3,300);
	VojenskeLetadlo *vl1 = new VojenskeLetadlo(10, VojenskeLetadlo::KONVENCNI);
	VojenskeLetadlo *vl2 = new VojenskeLetadlo (20, VojenskeLetadlo::ZADNE);
	VojenskeLetadlo *vl3 = new VojenskeLetadlo(30, VojenskeLetadlo::JADERNE);

	pole[0] = cl1;
	pole[1] = cl2;
	pole[2] = cl3;
	pole[3] = vl1;
	pole[4] = vl2;
	pole[5] = vl3;

	//vypsani pole
	for (int i = 0; i < SIZE; i++) {
		cout << pole[i]->ToString();
	}

	cout << "\n\n\n";

	//zaslani zpravy
	const char* zprava = "Rychlost vetru 50 km/h";
	for (int i = 0; i < SIZE; i++) {
		pole[i]->PosliZpravu(zprava);
	}

	system("pause");
	return 0;
}
```