

C++ - Quiz obiektowy

© Copyright by 3bird Projects 2024, <http://edukacja.3bird.pl>

Ogólne

Cały projekt jest podzielony na trzy pliki: **main.cpp**, **pytania.h** (czyli nagłówki metod), **pytania.cpp**. Aby włączyć te dwa dodatkowe pliki do projektu, należy w *Code::Blocks* kliknąć prawym przyciskiem myszy na drzewko z nazwą projektu i wydać polecenie „Add files...”. Dodatkowo należy także stworzyć czwarty plik z pytaniami (to będzie zwykły plik tekstowy): **quit.txt**. Ten plik musi w *Windows* mieć kodowanie znaków *cp852*, co możesz uzyskać np. za pomocą konwertera online: <http://www.motobit.com/util/charset-codepage-conversion.asp>

Uwaga: Nigdy nie wolno kopiować kodu z PDF-a, gdyż zawiera on niewidoczne znaki końca linii, twarde odstępki, odmienne cudzysłowy, odmienne apostrofy, odmienne myślniki. To godziny dodatkowej pracy na wykrywanie błędów i ich poprawianie. Kod należy przepisać ze zrozumieniem.

Kod - wersja podstawowa

Plik main.cpp

```
#include <iostream>
#include "pytania.h"
```

```
using namespace std;
```

```
int main() {
    int rozmiarTablicy = 5;
    SzablonPytania pytanie[rozmiarTablicy];
    int sumaPunktow = 0;

    cout << "Udziel odpowiedzi: A / B / C / D" << endl;
    for(int licznik = 0; licznik < rozmiarTablicy; licznik++) {
        pytanie[licznik].numerPytania = licznik+1;
        pytanie[licznik].wczytajPytaniazPliku();
        pytanie[licznik].wyswietlPytanie();
        pytanie[licznik].sprawdzOdpowiedz();
        sumaPunktow += pytanie[licznik].zdobytyPunkt;
    }

    cout << "\nZdobyta ilość punktów: " << sumaPunktow << endl;
    cout << "\nNaciśnij ENTER, aby zakończyć..." << endl;
    system("pause > nul");
    return 0;
}
```

Plik pytania.h

```
#include <iostream>
```

```
using namespace std;
```

```
class SzablonPytania {
```

```
public:
```

```
string trescPytania, odpowiedzA, odpowiedzB, odpowiedzC, odpowiedzD;
```

```
string odpowiedzPoprawna, odpowiedzUzytkownika;
```

```
int numerPytania, zdobytyPunkt;
```

```
void wczytajPytaniazPliku();
```

```
void wyswietlPytanie();
```

```
void sprawdzOdpowiedz();
```

```
};
```

Plik pytania.cpp

```
#include <iostream>
```

```
#include "pytania.h"
```

```
#include <fstream>
```

```
using namespace std;
```

```
void SzablonPytania::wczytajPytaniazPliku() {
```

```
    fstream plik;
```

```
    plik.open("quiz.txt", ios::in);
```

```
    if (plik.good()==false) {
```

```
        cout << "Jakiś problem z otwarciem pliku z pytaniami..." << endl;
```

```
        exit(0);
```

```
    }
```

```
    int numerLinii = ((numerPytania-1) * 7) + 1;
```

```
    int licznikPytania = 1;
```

```
    string biezacaLinia;
```

```
    while(getline(plik, biezacaLinia)) {
```

```
        if(licznikPytania==numerLinii) {
```

```
            trescPytania = biezacaLinia;
```

```
        }
```

```
        if(licznikPytania==numerLinii+1) {
```

```
            odpowiedzA = biezacaLinia;
```

```
        }
```

```
        if(licznikPytania==numerLinii+2) {
```

```
            odpowiedzB = biezacaLinia;
```

```
        }
```

```
        if(licznikPytania==numerLinii+3) {
```

```
            odpowiedzC = biezacaLinia;
```

```
        }
```

```

        if(licznikPytania==numerLinii+4) {
            odpowiedzD = biezacaLinia;
        }
        if(licznikPytania==numerLinii+5) {
            odpowiedzPoprawna = biezacaLinia;
        }
        licznikPytania++;
    }
    plik.close();
}

```

```

// =====
void SzablonPytania::wyswietlPytanie() {
    cout << endl << trescPytania << endl;
    cout << odpowiedzA << endl;
    cout << odpowiedzB << endl;
    cout << odpowiedzC << endl;
    cout << odpowiedzD << endl;
    cout << "-----" << endl;
    cout << "Odpowiedź: ";
    cin >> odpowiedzUzytkownika;
}

```

```

// =====
void SzablonPytania::sprawdzOdpowiedz() {
    if(odpowiedzUzytkownika == odpowiedzPoprawna) {
        zdobytyPunkt = 1;
    }
    else {
        zdobytyPunkt = 0;
    }
}

```

Plik quiz.txt

Uczniem Sokratesa był:

- a) Platon;
 - b) Kartezjusz;
 - c) Einstein;
 - d) Pitagoras;
- A

Stolicą Polski jest:

- a) Gdańsk;
 - b) Katowice;
 - c) Kraków;
 - d) Warszawa;
- D

Pierwiastkowanie jest odwrotnością:

- a) mnożenia;
- b) potęgowania;
- c) sumowania;
- d) odejmowania;

B

Polski sejm składa się z:

- a) 460 posłów;
- b) 100 posłów;
- c) 560 posłów;
- d) 400 posłów;

A

Najwyższy szczyt w Polsce to:

- a) Ślęża;
- b) Babia Góra;
- c) Rysy;
- d) Mont Blanc;

C

Polska ma dostęp do morza:

- a) Czarnego;
- b) Śródziemnego;
- c) Bałtyckiego;
- d) Północnego;

C

Kod - wersja rozbudowana (*Windows*)

Plik main.cpp

```
// Zbiór wszystkich cech obiektu to klasa. Klasa definiuje, czym ten dany obiekt jest.
// Klasa to odpowiednik gatunku w świecie zwierząt (nie konkretne zwierzę).
// Projekt domu - to klasa, a konkretny dom to obiekt tej klasy.
// W każdej klasie może być ta sama nazwa zmiennych (nie koliduje to ze sobą), a także ta
// sama nazwa funkcji (metod). Obiekty tej samej klasy mogą się od siebie różnić wartościami
// zmiennych (cech). W kodowaniu, klasa to własny typ danych, dlatego właśnie ten zapis:
// nazwaKlasy nazwaObiektu.
// nazwaObiektu.atrybut1 (atrybut to zmienna w klasie)
// nazwaObiektu.metoda() (metoda to funkcja w klasie)

#include <iostream> // Szukaj biblioteki w katalogu instalacyjnym Code::Blocks
#include <windows.h> // Potrzebne do aktywacji kolorów w konsoli
#include "pytania.h" // Szukaj biblioteki w katalogu projektu (dlatego cudzysłów).
// Uwaga: W przypadku innych edytorów, w powyższej linii wymagane może być "pytania.cpp"
```

```
using namespace std;
```

```
int main() {
```

```
    // ===== KOLOROWANIE NAPISÓW =====
    // Tworzymy tzw. uchwyt do tego, co będzie pojawiać się na konsoli (do bufora konsoli):
    HANDLE konsola = GetStdHandle(STD_OUTPUT_HANDLE);
    // Aktywujemy virtualny terminal i to, co będzie się na nim pojawiać:
    #ifndef ENABLE_VIRTUAL_TERMINAL_PROCESSING // Jeśli nie jest zdefiniowany, to:
    #define ENABLE_VIRTUAL_TERMINAL_PROCESSING 0x0004
    #endif
    // Aby powyższe działało, musimy aktywować i to, co poniżej:
    #ifndef ENABLE_PROCESSED_OUTPUT // Jeśli nie jest zdefiniowany, to:
    #define ENABLE_PROCESSED_OUTPUT 0x0001
    #endif

    // Wartość trybu (intup lub output). Słowo "dw" to skrót od "Display Window",
    // jest to jednak nazwa zmiennej, i może być inna:
    DWORD dwMode = 0;
    dwMode |= ENABLE_PROCESSED_OUTPUT | ENABLE_VIRTUAL_TERMINAL_PROCESSING;
    SetConsoleMode(konsola, dwMode);
    // ===== KOLOROWANIE NAPISÓW - KONIEC =====
```

```
    // W tym konkretnym projekcie, ustalamy odgórnie, ile pytań pojawi się w quizie:
```

```
    int rozmiarTablicy = 5;
```

```
    SzablonPytania pytanie[rozmiarTablicy]; // Tworzymy tablicę obiektów z ilością pytań
```

```
    int sumaPunktow = 0;
```

```
    cout << "\033[1;36;40m===== QUIZ =====\033[0m" << endl;
```

```
    cout << "\033[1;30;40mUdziel odpowiedzi: A / B / C / D\033[0m" << endl;
```

```
    for(int licznik = 0; licznik < rozmiarTablicy; licznik++) {
```

```
        pytanie[licznik].numerPytania = licznik+1;
        pytanie[licznik].wczytajPytaniazPliku();
        pytanie[licznik].wyswietlPytanie();
        pytanie[licznik].sprawdzOdpowiedz();
        sumaPunktow += pytanie[licznik].zdobytyPunkt;
```

```
    }
```

```
    cout << "\nZdobyta ilość punktów: \033[1;32;40m" << sumaPunktow << "\033[0m." << endl;
```

```
    cout << "\nNaciśnij ENTER, aby zakończyć..." << endl;
```

```
    system("pause > nul");
```

```
    return 0;
```

```
}
```

Plik pytania.h

```
// Pliki z rozszerzeniem *.h (od headers) są spisem treści zawierającym jedynie nagłówki metod.
```

```
// Same metody umieszczone są w pliku "pytania.cpp".
```

```
// Są tworzone głównie dla samych programistów.
```

```
// Aby dodać ten plik do projektu: PPM na nazwę projektu / Add files...
```

```
#include <iostream>
```

```
using namespace std;
```

```
class SzablonPytania {
```

```
    public:
```

```
    string trescPytania, odpowiedzA, odpowiedzB, odpowiedzC, odpowiedzD;
```

```
    string odpowiedzPoprawna, odpowiedzUzytkownika;
```

```
    int numerPytania, zdobytyPunkt;
```

```
    // Spis treści metod:
```

```
    void wczytajPytaniazPliku(); // Wczytuje pytania z zewnętrznego pliku i przypisuje je do  
    // zmiennych
```

```
    void wyswietlPytanie(); // Wyświetla pytanie i przypisuje odpowiedz użytkownika do  
    // zmiennej
```

```
    void sprawdzOdpowiedz();
```

```
};
```

```
Plik pytania.cpp
```

```
#include <iostream> // Szukaj biblioteki w katalogu instalacyjnym Code::Blocks (dlatego  
    // nawiasy trójkątne).
```

```
#include "pytania.h" // Szukaj biblioteki w katalogu projektu (dlatego cudzysłów)
```

```
#include <fstream> // Potrzebne do wczytania pliku tekstowego
```

```
#include <cctype> // Do użycia funkcji toupper lub tolower
```

```
using namespace std;
```

```
// W tym projekcie przyjmujemy, że pytania będzie pisać osoba nie będąca programistą.
```

```
// Stąd będą one wczytywane z pliku tekstowego
```

```
// Podwójny dwukropek to operator zasięgu, określa jakiej klasy będzie dotyczyć metoda.
```

```
// Stosowany, gdy definicja klasy umieszczona jest w innym pliku.
```

```
void SzablonPytania::wczytajPytaniazPliku() {
```

```
    fstream plik;
```

```
    plik.open("quiz.txt", ios::in); // Otwieramy plik do odczytu korzystając z metody "in" z klasy  
    // "ios"
```

```
    // Sprawdzamy za pomocą metody "good", czy plik został otwarty:
```

```
    if (plik.good()==false) {
```

```
        cout << "\033[1;31;40mJakiś problem z otwarciem pliku z pytaniami...\033[0m" << endl;  
        exit(0);
```

```
    }
```

```
    // Chcemy wczytać w każdym pytaniu 7 linii tekstu (razem z pustą linią oddzielającą).
```

```
    // Aby wiedzieć, od której linii zacząć czytać:
```

```
    int numerLinii = ((numerPytania-1) * 7) + 1; // Czyli w przypadku pytania nr 3: (3-1) * 7 + 1...
```

```
    // czyli od linii 15
```

```
    int licznikPytania = 1; // Które pytanie losujemy
```

```

string biezacaLinia;

// Getline wczytuje całe linie i zwraca prawdę dopóki istnieje linia do wczytania:
while(getline(plik, biezacaLinia)) {
    if(licznikPytania==numerLinii) {
        trescPytania = biezacaLinia;
    }
    if(licznikPytania==numerLinii+1) {
        odpowiedzA = biezacaLinia;
    }
    if(licznikPytania==numerLinii+2) {
        odpowiedzB = biezacaLinia;
    }
    if(licznikPytania==numerLinii+3) {
        odpowiedzC = biezacaLinia;
    }
    if(licznikPytania==numerLinii+4) {
        odpowiedzD = biezacaLinia;
    }
    if(licznikPytania==numerLinii+5) {
        odpowiedzPoprawna = biezacaLinia;
    }
    licznikPytania++;
}
plik.close();
}

// =====
void SzablonPytania::wyswietlPytanie() {
    cout << endl << "\033[1;33;40m" << trescPytania << "\033[0m" << endl;
    cout << odpowiedzA << endl;
    cout << odpowiedzB << endl;
    cout << odpowiedzC << endl;
    cout << odpowiedzD << endl;
    cout << "-----" << endl;
    cout << "Odpowiedź: ";
    cin >> odpowiedzUzytkownika;
    // Odpowiedź użytkownika może zawierać jeden znak lub wiele (gdy będzie złośliwy).
    // Może także zawierać znaki pisane małą literą lub dużą.
    // Dlatego wszystko to, co wpisał użytkownik zamieniamy na duże litery.
    // W poniższej pętli używamy referencji, aby bezpośrednio modyfikować elementy w stringu.
    // Bez referencji otrzymalibyśmy kopię każdego znaku (kopię w postaci tej zmiennej),
    // a zmiany w tej zmiennej nie wpływałyby na oryginalny string.
    for (char &kazdyZnakOdpowiedzi : odpowiedzUzytkownika) {
        kazdyZnakOdpowiedzi = toupper(kazdyZnakOdpowiedzi);
    }
}
}

```

```
// =====  
void SzablonPytania::sprawdzOdpowiedz() {  
    if(odpowiedzUzytkownika == odpowiedzPoprawna) {  
        zdobytyPunkt = 1;  
    }  
    else {  
        zdobytyPunkt = 0;  
    }  
}
```

Plik quiz.txt

Uczniem Sokratesa był:

- a) Platon;
- b) Kartezjusz;
- c) Einstein;
- d) Pitagoras;

A

Stolicą Polski jest:

- a) Gdańsk;
- b) Katowice;
- c) Kraków;
- d) Warszawa;

D

Pierwiastkowanie jest odwrotnością:

- a) mnożenia;
- b) potęgowania;
- c) sumowania;
- d) odejmowania;

B

Polski sejm składa się z:

- a) 460 posłów;
- b) 100 posłów;
- c) 560 posłów;
- d) 400 posłów;

A

Najwyższy szczyt w Polsce to:

- a) Ślęza;
- b) Babia Góra;
- c) Rysy;
- d) Mont Blanc;

C

Polska ma dostęp do morza:

- a) Czarnego;
 - b) Śródziemnego;
 - c) Bałtyckiego;
 - d) Północnego;
- C

Kod - wersja rozbudowana (*Linux*)

Plik main.cpp

```
// Zbiór wszystkich cech obiektu to klasa. Klasa definiuje, czym ten dany obiekt jest.  
// Klasa to odpowiednik gatunku w świecie zwierząt (nie konkretne zwierzę).  
// Projekt domu - to klasa, a konkretny dom to obiekt tej klasy.  
// W każdej klasie może być ta sama nazwa zmiennych (nie koliduje to ze sobą), a także ta  
// sama nazwa funkcji (metod). Obiekty tej samej klasy mogą się od siebie różnić wartościami  
// zmiennych (cech). W kodowaniu, klasa to własny typ danych, dlatego właśnie ten zapis:  
// nazwaKlasy nazwaObiektu.  
// nazwaObiektu.atrybut1 (atrybut to zmienna w klasie)  
// nazwaObiektu.metoda() (metoda to funkcja w klasie)
```

```
#include <iostream> // Szukaj biblioteki w katalogu instalacyjnym Code::Blocks  
#include "pytania.h" // Szukaj biblioteki w katalogu projektu (dlatego cudzysłów).  
// Uwaga: W przypadku innych edytorów, w powyższej linii wymagane może być "pytania.cpp"
```

```
using namespace std;
```

int main() {

```
    // W tym konkretnym projekcie, ustalamy odgórnie, ile pytań pojawi się w quizie:  
    int rozmiarTablicy = 5;  
    SzablonPytania pytanie[rozmiarTablicy]; // Tworzymy tablicę obiektów z ilością pytań  
    int sumaPunktow = 0;
```

```
    cout << "\n\033[1;36;40m===== QUIZ =====\033[0m" << endl;
```

```
    cout << "\033[1;30;40mUdziel odpowiedzi: A / B / C / D\033[0m" << endl;
```

```
    for(int licznik = 0; licznik < rozmiarTablicy; licznik++) {
```

```
        pytanie[licznik].numerPytania = licznik+1;  
        pytanie[licznik].wczytajPytaniazPliku();  
        pytanie[licznik].wyswietlPytanie();  
        pytanie[licznik].sprawdzOdpowiedz();  
        sumaPunktow += pytanie[licznik].zdobytyPunkt;
```

```
    }
```

```
    cout << "\nZdobyta ilość punktów: \033[1;32;40m" << sumaPunktow << "\033[0m." << endl;
```

```
    cout << "\nNaciśnij ENTER, aby zakończyć..." << endl;
```

```
    cin.ignore(); // Ignoruj wcześniejsze znaki ENTER...
```

```
    cin.get();
```

```
return 0;
}
```

```
// Info: Konwerter kodowania znaków:
// https://www.motobit.com/util/charset-codepage-conversion.asp
```

Plik pytania.h

```
// Pliki z rozszerzeniem *.h (od headers) są spisem treści zawierającym jedynie nagłówki metod.
// Same metody umieszczone są w pliku "pytania.cpp".
// Są tworzone głównie dla samych programistów.
// Aby dodać ten plik do projektu: PPM na nazwę projektu / Add files...
```

```
#include <iostream>
```

```
using namespace std;
```

class SzablonPytania {

```
public:
```

```
string trescPytania, odpowiedzA, odpowiedzB, odpowiedzC, odpowiedzD;
```

```
string odpowiedzPoprawna, odpowiedzUzytkownika;
```

```
int numerPytania, zdobytyPunkt;
```

```
// Spis treści metod:
```

```
void wczytajPytaniazPliku(); // Wczytuje pytania z zewnętrznego pliku i przypisuje je do
// zmiennych
```

```
void wyświetlPytanie(); // Wyświetla pytanie i przypisuje odpowiedz użytkownika do
// zmiennej
```

```
void sprawdzOdpowiedz();
```

```
};
```

Plik pytania.cpp

```
#include <iostream> // Szukaj biblioteki w katalogu instalacyjnym Code::Blocks (dlatego
// nawiasy trójkątne).
```

```
#include "pytania.h" // Szukaj biblioteki w katalogu projektu (dlatego cudzysłów)
```

```
#include <fstream> // Potrzebne do wczytania pliku tekstowego
```

```
#include <cctype> // Do użycia funkcji toupper lub tolower
```

```
using namespace std;
```

```
// W tym projekcie przyjmujemy, że pytania będzie pisać osoba nie będąca programistą.
```

```
// Stąd będą one wczytywane z pliku tekstowego
```

```
// Podwójny dwukropek to operator zasięgu, określa jakiej klasy będzie dotyczyć metoda.
```

```
// Stosowany, gdy definicja klasy umieszczona jest w innym pliku.
```

```
void SzablonPytania::wczytajPytaniazPliku() {
```

```
    fstream plik;
```

```
    plik.open("quiz.txt", ios::in); // Otwieramy plik do odczytu korzystając z metody "in" z klasy
```

```

// "ios"

// Sprawdzamy za pomocą metody "good", czy plik został otwarty:
if (plik.good()==false) {
    cout << "\033[1;31;40mJakiś problem z otwarciem pliku z pytaniami...\033[0m" << endl;
    exit(0);
}

// Chcemy wczytać w każdym pytaniu 7 linii tekstu (razem z pustą linią oddzielającą).
// Aby wiedzieć, od której linii zacząć czytać:
int numerLinii = ((numerPytania-1) * 7) + 1; // Czyli w przypadku pytania nr 3: (3-1) * 7 + 1...
// czyli od linii 15
int licznikPytania = 1; // Które pytanie losujemy
string biezacaLinia;

// Getline wczytuje całe linie i zwraca prawdę dopóki istnieje linia do wczytania:
while(getline(plik, biezacaLinia)) {
    if(licznikPytania==numerLinii) {
        trescPytania = biezacaLinia;
    }
    if(licznikPytania==numerLinii+1) {
        odpowiedzA = biezacaLinia;
    }
    if(licznikPytania==numerLinii+2) {
        odpowiedzB = biezacaLinia;
    }
    if(licznikPytania==numerLinii+3) {
        odpowiedzC = biezacaLinia;
    }
    if(licznikPytania==numerLinii+4) {
        odpowiedzD = biezacaLinia;
    }
    if(licznikPytania==numerLinii+5) {
        odpowiedzPoprawna = biezacaLinia;
    }
    licznikPytania++;
}
plik.close();
}

// =====
void SzablonPytania::wyswietlPytanie() {
    cout << endl << "\033[1;33;40m" << trescPytania << "\033[0m" << endl;
    cout << odpowiedzA << endl;
    cout << odpowiedzB << endl;
    cout << odpowiedzC << endl;
    cout << odpowiedzD << endl;
}

```

```

cout << "-----" << endl;
cout << "Odpowiedź: ";
cin >> odpowiedzUzytkownika;
// Odpowiedź użytkownika może zawierać jeden znak lub wiele (gdy będzie złośliwy).
// Może także zawierać znaki pisane małą literą lub dużą.
// Dlatego wszystko to, co wpisał użytkownik zamieniamy na duże litery.
// W poniższej pętli używamy referencji, aby bezpośrednio modyfikować elementy w stringu.
// Bez referencji otrzymalibyśmy kopię każdego znaku (kopię w postaci tej zmiennej),
// a zmiany w tej zmiennej nie wpływałyby na oryginalny string.
for (char &kazdyZnakOdpowiedzi : odpowiedzUzytkownika ) {
    kazdyZnakOdpowiedzi = toupper(kazdyZnakOdpowiedzi);
}

// =====
void SzablonPytania::sprawdzOdpowiedz() {
    if(odpowiedzUzytkownika == odpowiedzPoprawna) {
        zdobytyPunkt = 1;
    }
    else {
        zdobytyPunkt = 0;
    }
}

```

Plik quiz.txt

Uczniem Sokratesa był:

- a) Platon;
 - b) Kartezjusz;
 - c) Einstein;
 - d) Pitagoras;
- A

Stolicą Polski jest:

- a) Gdańsk;
 - b) Katowice;
 - c) Kraków;
 - d) Warszawa;
- D

Pierwiastkowanie jest odwrotnością:

- a) mnożenia;
 - b) potęgowania;
 - c) sumowania;
 - d) odejmowania;
- B

Polski sejm składa się z:

- a) 460 posłów;
- b) 100 posłów;
- c) 560 posłów;
- d) 400 posłów;

A

Najwyższy szczyt w Polsce to:

- a) Ślęża;
- b) Babia Góra;
- c) Rysy;
- d) Mont Blanc;

C

Polska ma dostęp do morza:

- a) Czarnego;
- b) Śródziemnego;
- c) Bałtyckiego;
- d) Północnego;

C

Ostatnia aktualizacja: 25 stycznia 2024.