Programowanie obiektowe C++

Obiekty, tablice obiektów

Obiekty, tablice obiektów

1. Przykład

Poniższy przykład ilustruje pracę z obiektami opisującymi studentów.

```
#include <iostream>
#include <string>

using namespace std;

class TStudent
{
  public:
    TStudent(string nazwisko, string imie, int nrIndeksu, int nrRoku);

    string imieNazwisko();
    int sprNrIndeksu();

  private:
    string nazwisko;
    string imie;
    int nrIndeksu;
    int nrRoku;
    int liczbaPrzed;
    string przedmioty[16];
};

TStudent::TStudent(string nazwisko, string imie, int nrIndeksu, int nrRoku)
: nazwisko(nazwisko), imie(imie), nrIndeksu(nrIndeksu), nrRoku(nrRoku)
{
};

string TStudent::imieNazwisko()
{
  string str = imie + " " + nazwisko;

  return str;
};

int TStudent::sprNrIndeksu()
{
  return nrIndeksu;
};

int main()
{
  TStudent stud("Kowalski", "Albert", 1, 1);
```

```
  cout << stud.imieNazwisko( ) << endl;
  cout << stud.sprNrIndeksu( ) << endl;

  return 0;
}
```

Wprowadź program i przetestuj jego działanie dla przykładowych danych.

2. Zadanie

Uzupełnij powyższy kod o następujące metody:

- konstruktor pozwalający wpisywać nazwy przedmiotów do tablicy **`przedmioty[]`**,
- **`sprWszystko()`** – zwraca łańcuch znaków ze wszystkimi informacjami o studencie (włącznie z nazwami przedmiotów); bez argumentów,
- **`ustawNrIndeksu()`** – pozwala na zmianę numeru indeksu studenta; argumentem jest nowy numer indeksu, zwraca **`void`**,
- **`sprRok()`** – pozwala na sprawdzenie roku studiów studenta; bez argumentów, zwraca numer roku,
- **`zwiekszRok()`** – inkrementuje rok studiów o jeden; zwraca **`void`**,
- **`dodajPrzedmiot()`** – dodaje nazwę przedmiotu do tablicy, zwraca **`void`**,
- **`usunPrzedmiot()`** – usuwa przedmiot i na jego miejsce przenosi ostatni przedmiot w tablicy; zwraca **`void`**.

Do napisania powyższych funkcji może być przydatna przeciążona funkcja **`std::to_string()`**, zamieniająca wartości różnych typów na łańcuch znaków oraz zdefiniowany w przestrzeni nazw **`std`** przeciążony operator relacji **`==`**, który można stosować z łańcuchami znaków.

Zadania dodatkowe

1. Program z deklarację obiektu **`TKwadrat`**:

```
class TKwadrat
{
  public:
    float obwod( );
    float pole( )

  private:
    float bok;
};
```

Uzupełnij o:

- odpowiednie zadeklarowane metody,
- metody pozwalające na odczyt i zmianę boku kwadratu,
- domyślny konstruktor klasy (bez parametrów lub z domyślnymi parametrami),
- zdefiniuj tablicę obiektów (np.: **`TKwadrat kwadraty[5]`**) i zmień niektóre wielkości boków kwadratów,
- wykonaj obliczenia pola i obwodu po wszystkich elementach tablicy w pętli **`for()`**.