

Obiekty, tablice obiektów

1. Przykład

Poniższy przykład ilustruje pracę z obiektami opisującymi studentów.

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

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

    std::string imieNazwisko();
    int sprNrIndeksu();

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

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

std::string TStudent::imieNazwisko()
{
    std::string str;

    str.append(imie);
    str.append(" ");
    str.append(nazwisko);

    return str;
};

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

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

    std::cout << stud.imieNazwisko() << std::endl;
    std::cout << stud.sprNrIndeksu() << std::endl;

    return EXIT_SUCCESS;
};
```

```
| }
```

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,
- **zwiększRok()** – 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_str()**, 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 deklaracją 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()**.