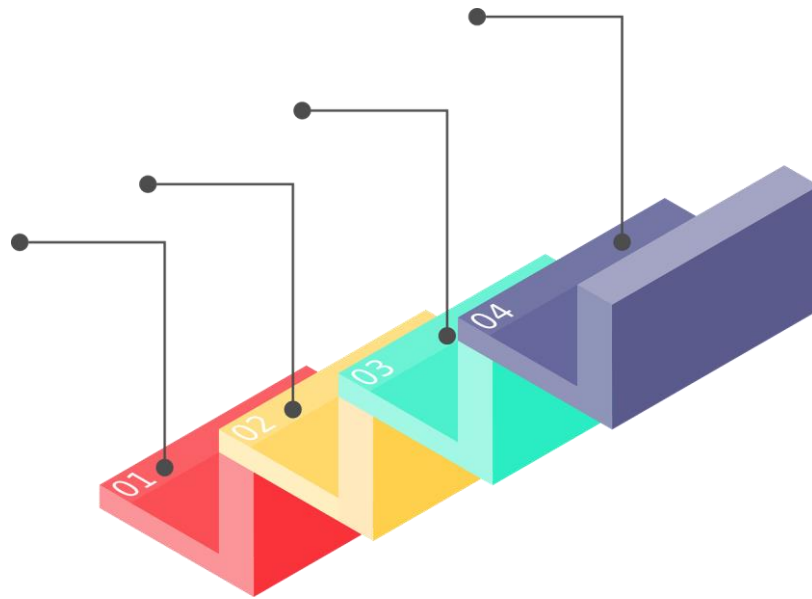



Podsumowanie – struktury i klasy




Moją uwagę zwróciło kilka szczegółów w Państwa kodach:

-  Wskaźnik do obiektu klasy będzie **typu tej klasy** (wskaźnik musi być tego samego typu, co obiekt na który ma wskazywać):

```
class T_student
{
public:
    //definicja składników klasy
};

int main()
{
    T_student student1; //definicja obiektu student1 typu klasy T_student
    T_student student2;
    T_student* wsk = &student2; //definicja wskaźnika, który będzie wskazywał na
    obiekt student2 klasy T_student
}
```

-  Tworząc obiekty danej klasy nie musimy podawać (w języku C++) słowa kluczowego *struct/class*:

```


struct T_osoba
{
    //definicja składników klasy
};

int main()
{

    //struct T_osoba student1;
    //zapis z języka C, gdzie przy tworzeniu obiektu struktury trzeba było zapisać
    słowo kluczowe struct; kompilator C++ nie wyrzuci błędu

    T_osoba student2; //stworzenie obiektu klasy T_osoba bez konieczności powtarzania
                        //kompilatorowi informacji, że jest to obiekt klasy
}

```

- 
 Zagnieżdżenie obiektu jednej klasy w drugiej, to nic innego jak dodanie w jednej klasie obiektu o typie drugiej klasy:

```

class T_student
{
private:

    string nazwisko;
    int rok;
    T_data urodziny;

public:

    void Wypisz();

};

void T_student::Wypisz()
{

    cout <<"Inicjal: " << inicjal << "." << "Nazwisko: " << nazwisko << "," <<"Rok: "
    << rok << "," << "Grupa: " << grupa << endl;
    urodziny.Wypisz_date(); //wykorzystanie funkcji składowej klasy T_student do
                            //wypisania danych składnika urodziny o typie klasy
                            //T_data
}

```