

WZORCE = SZABLONY

Po co używamy?

SPOSÓB DEKLARACJI

```
template <typename T, typename M>  
class Klasa {  
    // tu uzywamy typow T i M  
};
```

```
Klasa<double,int> x;
```

```
Klasa<int,Osoba> z;
```

SPOSÓB DEKLARACJI

```
template <typename T, int size>  
class Klasa {  
    // w definicji uzywamy typu 'T'  
    // i wartosci calkowitej 'size'  
};
```

```
Klasa<Osoba,100> t;
```


DEFINIOWANIE POZA SZABLONEM

```
template <typename T, int size>
class Klasa {
    void metoda1() {
        // definicja metody1
    }
    T* metoda2(double); // tylko deklaracja

    // ...
};

//
// ...
//

// definicja metody metoda2
template <typename T, int size>
T* Klasa<T,size>::metoda2(double x) {
    // ciało definicji
}
```

WZORCE KLAS POCHODNYCH

```
template<typename Typ>
class Klasa {
public:
    int size
    Klasa(int s) :size(s) {};
};
```

```
template<typename Typ>
class Klasa_pochodna : public Klasa<Typ> {
    Typ dodatkowe_pole
public:
    Klasa_pochodna(int s, Typ dodp) :Klasa<Typ>(s), dodatkowe_pole(dodp) {};
};
```

WZORZEC SZCZEGÓŁOWY

```
template <typename T>
class Klasa {
    // szablon podstawowy
};

template <>
class Klasa<char *> {
    // szablon szczegółowy dla typu char *
};
```

ZMENNA LICZBA ARGUMENTÓW

```
template <typename T>
void Funkcja(T wartosc) {
    // Działanie dla ostatniego elementu
}
```

```
template <typename Pierwszy, typename ... Reszta>
void Funkcja(Pierwszy wartosc1, Reszta ... wartoscn) {
    // Działanie dla pozostałych elementów
    Funkcja(wartoscn...);
}
```


ALIASY

```
template<typename T>  
using Alias = Klasa<T, 20>;  
  
Alias<int> nowy;
```


EXTERN

```
// header.h

template<typename T>
void Funkcja()
{
    // Ciało funkcji
}
```

```
// source1.cpp

#include "header.h"
void FunkcjaNadrzedna1()
{
    Funkcja<int>();
}
```

```
// source2.cpp

#include "header.h"
extern template void Funkcja<int>();

void FunkcjaNadrzedna2()
{
    Funkcja<int>();
}
```