* Шаблони на функции

template <typename T>

void read(int n, T\* a) {

for (int i = 0; i < n; i++) {

cout << "a[" << i << "]= ";

cin >> a[i];

}

}

* Шаблони на класове – позволява създаването на класове, използващи неопределени типове данни за свои полета, параметри на методи и връщан резултат от методи.

Такива класове са наричат обобщени.

Използват се за изграждане на общоцелеви класове – контейнери (стекове, опашки, списъци, вектори, дървета)

* Декларация на шаблон на класа

template <typename T, typename S = int>

class example {

public:

T func1(T, S);

S func2(T, S);

private:

T a;

S b;

}

* Дефиниране на методите на шаблонен клас

1. Вътрешни

public: T func1(T x, S y) {// вградена член-функция

cout << "func1 \n"; return x;

}

1. Външни

template <typename T, typename S>

S example<T, S>::func2(T x, S y) {

cout << "func2\n"; return y;

}

* Специализация на методите за съответния тип

template<>

void stack<char>::print() {

char x;

while (!empty()) {

pop(x);

cout << (int)x << " ";

} cout << endl;

}

* Приятелски функции и класове

class A {

friend class B;

}

* Наследяване при шаблонни класове

1. Базовият само е шаблонен
2. И базовият и производният са шаблонни – 2 варианта