- Что такое шаблонный класс?

Класс с шаблонными полями и методами.

- Для чего нужны шаблонные классы?

Шаблоны позволяют определить конструкции (функции, классы), которые используют определенные типы, но на момент написания кода точно не известно, что это будут за типы. Иными словами, шаблоны позволяют определить универсальные конструкции, которые не зависят от определенного типа.
Шаблон класса (class template) позволяет задать тип для объектов, используемых в классе

- В каких случая надо использовать Stack, а в каких Stack<T>при работе с шаблонным классом Stack. В чем разница?

1) Stack можно использовать в тех местах, где при объявлении требуется только имя, а не тип класса. (в частности, для конструкторов и деструкторов). Можно также использовать, когда компилятор может понять с каким типом инициализируется объект класса.
2) Stack<T> используется в местах с особой обработкой параметров. Также такая запись необходима вне структуры класса.

- Как инстанцируются шаблонные функции-члены класса? Что такое частичное использование шаблона. Приведите свои примеры

Заметим, что инстанцирование происходит только для вызываемых функций(членов) шаблона. Для шаблонов классов функции-члены инстанцируются только при их использовании. Очевидно, что такой подход позволяет экономить время, память и использовать шаблоны классов только частично.
Частичное использование шаблона - случай когда не все методы могут испольняться при текущем инстанцированном параметре.

```
class JustInt
{
public:
    int a;
    JustInt(int temp) {
        a = temp;
    }
    void doSmth() {
        std::cout << a;
    }
};

template<class T>
class ClassTemplate {
public:
    T elem;
    ClassTemplate(T val) {
        elem = val;
    }
    void tmplDoSmth() {
        std::cout << "Hello, world!";
    }
    void tmplDo2() {
        JustInt test(this.elem);
        test.doSmth();
    }
};
```

```cpp
int main() {
    ClassTemplate<int> a(2);
    a.tmplDoSmth();
    a.tmplDo2(); // OK


    ClassTemplate<std::string[]> b(new string{ "str1", "str2" });
    b.tmplDoSmth();
    b.tmplDo2(); // should error
}
```