

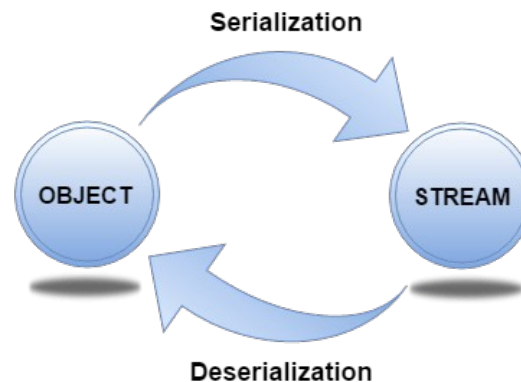
Boost serialization

Понятие сериализации

- Wiki: In computing, serialization is the process of **translating a data structure or object state into a format** that can be stored or transmitted and reconstructed later.

Форматы:

- 1) non-binary: json, xml, txt, yml, ...
- 2) binary
- 3) user-defined



Boost

```
template<class T> polymorphic_oarchive & operator<<(T & t){  
    return polymorphic_oarchive::operator<<(t);  
}
```

Boost

```
template<class T> polymorphic_oarchive & operator<<(T & t){  
    return polymorphic_oarchive::operator<<(t);  
}
```

```
template<class T> polymorphic_oarchive & operator&(T & t){  
    return polymorphic_oarchive::operator&(t);  
}
```

Boost

```
template<class T> polymorphic_oarchive & operator<<(T & t){  
    return polymorphic_oarchive::operator<<(t);  
}
```

```
template<class T> polymorphic_oarchive & operator&(T & t){  
    return polymorphic_oarchive::operator&(t);  
}
```

Передача *реальной* сериализации - дочернему типу T.

- <<, >>: операторы записи/чтения в/из архив(a).
- &: оператор сериализации.

```
ar << data;  
ar & data;
```

```
ar >> data;  
ar & data;
```

friend boost::serialization::access

- При переопределении шаблона функции

```
template<class Archive>  
void serialize(Archive& ar, [[maybe_unused]]  
               const unsigned int version);
```

в ней можно вызывать оператор &.

- Рекурсивный поиск данной функции в нетривиальных сериализуемых типах.
- `[[maybe_unused]]` для флага `-Wextra`