

1) Почему класс `std::string` имеет много перегруженных функций-членов?

`std::string` поддерживает работу одновременно с себе подобными (`std::string`) и классическими C-строками, а также одновременно с индексами и итераторами, что требует множества перегрузок.

2) Как осуществляется интернационализация и локализация программы?

Операции с символами осуществляются средствами `boost`, настройка потоков ввода-вывода под местные форматы и правила производится с помощью `facet`-ов из `std::locale`

3) Чем отличаются многобайтовые кодировки от широких кодировок?

Многобайтовые кодировки в зависимости от символа могут использовать разное число байт для хранения его кода, в то время как в широких кодировках для всех символов размер их кодов одинаков.

4) Какие компоненты входят в стандарт кодирования символов Unicode?

а) UCS – общая таблица кодов символов

б) UTF – набор кодировок, задающих соответствие между кодами символов и числами (битовыми представлениями), которые хранятся в памяти

5) Для решения каких задач удобно использовать регулярные выражения?

Регулярные выражения удобно использовать для поиска фрагментов текста, соответствующих или несоответствующих некоторому образцу (в точности или по формату) и, возможно, их редактирования.