

1. Какие концепции лежат в основе стандартной библиотеки?

- 1) Переносимость (одинаковая работа при, возможно, различных реализациях для разных систем)
- 2) Компактность (реализуются достаточно эффективные алгоритмы, но ради мелких оптимизаций громоздкий код не пишется)
- 3) Связывание (классы стандартной библиотеки активно используются программистами для согласования своих фрагментов кода (например, один из них написал некий алгоритм, вывел vector, другой обрабатывает готовый vector от первого программиста))

2. Зачем в проектах используются системы контроля версий?

Системы контроля версий нужны для эффективной работы по обновлению кода: можно подгружать в общий репозиторий свои изменения, скачивать измененные версии других программистов, разделять ветки (например, на стабильную, находящуюся в разработке и тестируемую пользователями версии), склеивать изменения из разных веток (модифицировать их структуру), делать откаты плохих обновлений и т.д.

3. Из каких основных действий состоит взаимодействие с Git?

- 1) clone — скачивание репозитория
- 2) commit — сохранение изменений в локальном репозитории
- 3) discard — отменяет изменения, не попавшие в commit
- 4) revert — делает commit, нейтрализующий изменения других commit-ов
- 5) reset — делает откат commit-ов, как будто их никогда не было
- 6) push — отправляет свежие commit-ы с локального репозитория в удаленный
- 7) pull — подгружает удаленный репозиторий и встраивает изменения с него в локальный (по принципу merge)
- 8) merge — добавляет изменения одной ветки в другую, сохраняя историю
- 9) rebase — добавляет изменения одной ветки в другую таким образом, чтобы они исторически шли до расхождения веток.

4. Когда следует создавать отдельные ветки для разработки?

Создание отдельных веток имеет смысл при наличии стабильной версии (возможно, еще не законченного продукта), чтобы можно было безопасно выполнить изменение кода, отладку и тестирование новой версии, а после этого встроить новые commit-ы, зная, что все шаги были удачными.

5. Какие основные элементы содержатся в библиотеке chrono?

Библиотека chrono содержит классы duration (интервал времени, задаваемый числом тактов и их продолжительностью в секундах, дополнительно прописаны псевдонимы для разных единиц), time_point (задает момент времени в виде интервала, прошедшего с некоторой эпохи, являющейся для него началом отсчета) и структуры system_clock, steady_clock (часы, задающие минимальный такт, эпоху и возможность/невозможность перевода времени).