

# Контрольные вопросы. Задание 12.

6 февраля 2023 г.

## 1 Какие концепции лежат в основе стандартной библиотеки?

Концепции, лежащие в основе стандартной библиотеки:

- переносимость;
- компактность/эффективность;
- связывание.

При этом реализации стандартной библиотеки могут отличаться, но они всегда обеспечивают одинаковое поведение.

## 2 Зачем в проектах используются системы контроля версий?

Системы контроля версий позволяют удобным, логичным и контролируемым образом (можно контролировать историю разработки, права доступа, различные версии) обмениваться рабочими материалами (программами, файлами) при реализации того или иного проекта, поэтому они и используются, причём как в одиночных проектах, так и в более крупных.

### 3 Из каких основных действий состоит взаимодействие с git?

Основные действия с git:

- clone – создание копии репозитория.
- commit – сохранение последних изменений файла в репозитории.
- push – загрузка содержимого локального репозитория в удалённый.
- pull – загрузка содержимого удалённого репозитория в локальный. При этом любые внешние коммиты сливаются в ветку, в которой сейчас работает разработчик.

Последние две команды обеспечивают синхронизацию содержимого локального и удалённого репозитория. Также можно выделить команды fetch, merge и rebase. Первая из них собирает все коммиты из целевой ветки, которых нет в текущей ветке, и сохраняет их в локальном репозитории, не объединяя их в текущей ветку. Последние отвечают за слияние ветвей разработки.

## 4 Когда следует создавать отдельные ветки для разработки?

Отдельные ветки для разработки следует создавать для параллельной разработки нескольких версий программы (контроль версий). Например, одна версия может быть стабильной для работы с клиентами, в ней будут лишь исправляться ошибки. При этом параллельные ветви могут быть использованы для улучшения каких-либо характеристик программы, для грядущих изменений (после их отладки).

## 5 Какие основные элементы содержатся в библиотеке chrono?

Основные элементы библиотеки chrono:

- clock (таймер) – тип, определяющий эпоху и продолжительность такта.
- duration (интервал) – тип, определяющий промежуток времени, состоящий из некоторого количества тактов заданной продолжительности.
- timepoint (момент времени) – тип, определяющий продолжительность времени, прошедшего с начала отсчёта (эпохи). Иными словами, это комбинация эпохи и соответствующего интервала времени (отсчитываемого от начала отсчёта).

## **Литература.**

[1] Конспект семинара. Макаров И.С.

[2] <https://en.cppreference.com/w/cpp/chrono>.

[3] <https://tproger.ru/explain/git-pull-and-git-fetch-whats-the-difference/>