Какие существуют способы обработки различных ошибок?

- Обработка “на месте”:
    1. **if-then-else** для проверки пользовательского вводы;
    2. **abort-exit-terminate** аварийное завершение работы;
    3. **assert** разыменование nullptr, ошибка программиста.
- Передача информации об ошибке (уведомление):
    1. коды возврата, либо доп.аргумент либо возвращаем структуру. Ну или просто возвращение номера ошибки, если такое возможно
    2. механизм исключений

В чём заключаются недостатки механизма кодов возврата?

Неудобство

1) плохая читабельность (можно использовать перечисления)
2) неудобно поддерживать, расширять и развивать
3) проблема в случае, если функция может вернуть любое значение из всей числовой прямой
4) не сочетаются в конструкторами, деструкторами и операторами

Какими особенностями обладает механизм исключений?

- многоуровневость
- расширяемость
- понятность

throw - выбрасывание исключения в область программы, откуда функция была вызвана

try - область захвата выброшенных исключений

catch - обработка исключений заданного типа или всех

Для чего используется спецификатор и оператор noexcept?

- спецификатор времени компиляции - позволяет сгенерировать более оптимальный код
- оператор - функция является noexcept согласно условию

Как формулируются гарантии безопасности исключений?

- базовая гарантия (RAII)
    1. инварианты не нарушены
    2. нет утечек ресурсов
- строгая гарантия (банк)
    1. транзакционное поведение
- отсутствие исключений (swap)
    1. исключения не генерируются