Программирование на языке C++

Шаблов Анатолий
anatoliishablov@gmail.com

ИТМО, весенний семестр 2023

Инкапсуляция, наследование, полиморфизм

Инкапсуляция

- Сокрытие логически связанных подробностей в одном месте.
- Объединение данных и логики их обработки.

Помогает разделять программу на слабо связанные компоненты. Варианты:

- ООП.
- Пространства имён и модули (Erlang, Haskell).
- Функции с локальным контекстом (Javascript, Scheme, Pascal).

Зачем нужно разделение?

- Легче читать.
- Легче менять.
- Легче тестировать.

Наследование

- Возможность переиспользовать код.
- Возможность расширять функциональность.
- Возможность компоновать код.

Статический и динамический полиморфизм

Статический:

- Использование одного кода для разных типов.
- На этапе компиляции полиморфизм раскрывается в обычный код.
- На этапе исполнения не остается следов.

Динамический:

- Возможность работать с объектами разных типов через один интерфейс.
- Поведение выбирается на этапе исполнения.
- Рефлексия.
  Гетерогенные коллекции.

Динамический полиморфизм в C++

- Наследование классов.
- Ссылки или указатели.
- Приведение типов.
- Виртуальные методы.

Виртуальные методы классов

Виртуальные методы

```
virtual foo();
```

Ограничения применения виртуальности

Не могут быть виртуальными:

- Свободные функции.
- Статические методы.
- Шаблонные методы.

Переопределение виртуальных методов

Если класс ниже по иерархии наследования определяет метод

- С тем же именем.
- С тем же списком параметров.
- С теми же cv-квалификаторами.
- С теми же ссылочными квалификаторами.

то этот метод тоже становится виртуальным и переопределяет исходный.

Ковариантный тип

Y ковариантен X, если

- Оба являются ссылкой или указателем на класс
- Y не более cv-квалифицирован, нежели X
- Класс, на который ссылается/указывает X, должен быть однозначным и доступным предком класса, на который ссылается/указывает Y

override и final

Спецификаторы override и final можно использовать в конце объявления метода.

override – метод переопределяет унаследованный виртуальный метод.

final – метод переопределяет унаследованный виртуальный метод и не может быть переопределён в наследниках.

Рекомендации по использованию виртуальности

- Интерфейс (в базовом классе) делать не виртуальным.
- Переопределяемую реализацию оформлять приватными виртуальными методами базового класса.
- Методы интерфейса вызывают нужные виртуальные методы реализации.
- Если предполагается полиморфное создание/удаление - сделать виртуальный деструктор в базовом классе.
- Иначе рассмотреть возможность защищённого не виртуального деструктора.

Технические аспекты наследования

Приведение типов по иерархии наследования

Можно приводить указатели и ссылки на классы вверх и вниз по иерархии наследования.

- Неявное приведение вверх: Derived * →Base *.
- Явное приведение вниз: Base * →Derived *.
- Типо-безопасное приведение по иерархии наследования.

При таком приведении сам объект не меняется – меняется лишь указатель или ссылка для доступа к нему.

Виртуальное наследование

При указании наследования используется ключевое слово virtual

- В каждом конкретном типе-наследнике виртуальный предок встречается как подобъект ровно один раз.
- Конкретный тип-наследник создаёт подобъекты виртуальных предков (даже если в иерархии наследования между ними другие классы), причём до создания подобъектов обычных предков.
- Наследник должен иметь доступ к конструктору виртуального предка.
- В списке инициализации конструктора наследника можно явно вызвать конструктор виртуального предка.
- Операция копирования конкретного наследника должна учитывать эти особенности.
- Для приведения типов по иерархии наследования с виртуальным наследованием необходимо использовать dynamic_cast.