

Принципы проектирования

Интерфейс - forward declarations

Контекст - реализация классов, макросы. Контекст позволяет выделить и сохранять инварианты объектов.

Инварианты - утв-я, которые выполняются на протяжении жизни объекта --> должны быть защищены и отделены от внешнего мира с помощью API --> не требуют проверки на каждом шаге.

Производный контекст физически расширяет базовый (а логически - наоборот)

Полиморфизм - разное поведение в зависимости от контекста. Например: шаблонные функции, виртуальные функции(динамический полиморфизм по неявному аргументу `this`), перегрузки.

Артефакт - (суп как результат приготовления супа)

Результатом проектирования является задание **отношения между типами данных**.

Виды отношений:

- А наследует от Б
- А является полем Б
- А находится в одном физическом модуле с Б
- А является аргументом метода Б
- другие

Результат проектирования - граф с вершинами (классами) и ребрами (отношения)

Для этого создан UML.

Класс представлен 3мя квадрантами: имя, поля, методы, другие атрибуты.

Отношения UML

Ассоциация

Сущности каким-то образом связаны друг с другом (например, появляются вместе внутри одной функции)