

Министерство образования Республики Беларусь
Учреждение образования «Белорусский государственный университет
информатики и радиоэлектроники»
Институт информационных технологий
Факультет компьютерных технологий
Кафедра информационных систем и технологий

Контрольная работа
по дисциплине
«Основы программной инженерии»

Выполнил студент группы 181073
Михолап Александр Владимирович
Проверил
Матвеев Андрей Владимирович

Минск 2021

Задание 1.

22. Ковариантность и контравариантность типов данных в языках программирования.

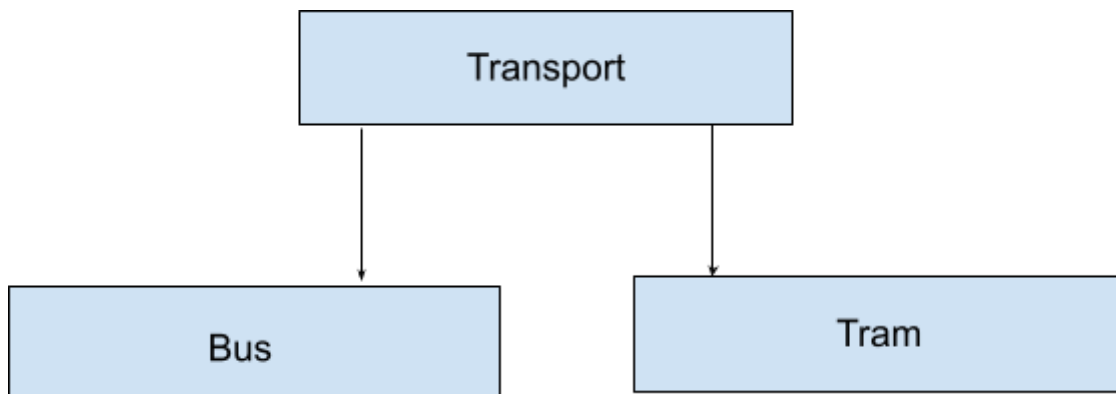
Вариантность — перенос наследования исходных типов на производные от них типы. Под производными типами понимаются контейнеры, делегаты, обобщения, а не типы, связанные отношениями "предок-потомок".

Ковариантность — перенос наследования исходных типов на производные от них типы в прямом порядке.

Контравариантность — перенос наследования исходных типов на производные от них типы в обратном порядке.

Если у производных типов наблюдается ковариантность, говорят, что они ковариантны исходному типу. Если у производных типов наблюдается контравариантность, говорят, что они контравариантны исходному типу.

В виде примера возьмем следующую иерархию типов.



При ковариантности тип Transport мы сможем поместить объект Bus или Tram, но не наоборот.

А вот при контравариантности наоборот, мы можем поместить в Bus или Tram объект Transport

Задание 2.

22. Сервис-ориентированная архитектура, ее преимущества и недостатки.

Сервис-ориентированная архитектура (service-oriented architecture, SOA) придумана в конце 1980-х.

SOA — это набор архитектурных принципов, не зависящих от технологий и продуктов, совсем как полиморфизм или инкапсуляция.

Достоинства:

1. Независимость от выбранных технологий.
2. Независимость от особенностей передачи данных/связи.

Недостатки:

1. Независимость от местоположения: клиентский код не имеет понятия, является ли вызов локальным или удаленным. Звучит неплохо, но длительность задержки и виды сбоев могут сильно варьироваться. Если мы не знаем, какой у нас вызов, то приложение не может выбрать подходящую стратегию обработки вызовов методов, а значит, и генерировать удаленные вызовы внутри цикла. В результате вся система работает медленнее.

2. Сложная, раздутая и неоднозначная спецификация: её собрали из нескольких версий спецификаций разных вендоров, поэтому (на тот момент) она была раздутой, неоднозначной и трудной в реализации.

3. Заблокированные каналы связи (communication pipes): используются специфические протоколы поверх TCP/IP, а также специфические порты (или даже случайные порты). Но правила корпоративной безопасности и фаерволы зачастую допускают HTTP-соединения только через 80-й порт, блокируя обмены данными CORBA.