

Люк,
я твой
отец



ООП/С++: Лекция 3

Наследование

ака «Последнее главное слово»

<https://github.com/avasyukov/oop-2nd-term/tree/master/2019/lection03>



http://judge2.vdi.mipt.ru/cgi-bin/new-client?contest_id=911139

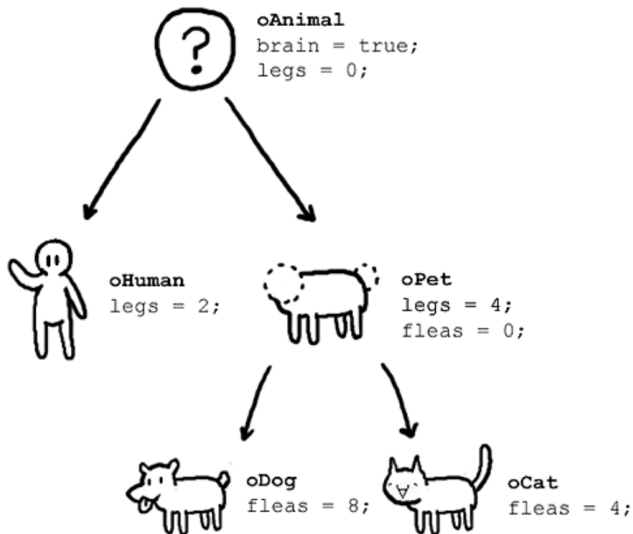


Наследование

Наследование: о чём это

Всё наследование – это о том, как не писать повторно один и тот же код для «почти одинаковых» сущностей.

Наследование: о чём это



Наследование: перегрузка

При наследовании от класса вы можете получить:

- Его собственные поля и методы.
- В неизменном виде то, что он сам унаследовал ранее.
- Его унаследованные, а потом творчески перегруженные методы.

Наследование: перегрузка



Наследование: примеры

Разбираем первый пример кода к данной лекции

- 01_animals.cpp

Виртуальные функции

Иногда в родительском классе можно сказать только «здесь должен быть логически вот такой метод», но нельзя написать его реализацию.

- Заведомо предполагается, что будут унаследованные классы.
- Метод для них всех нужен, можно логически сказать, что метод должен делать.
- Реализация будет кардинально разной в разных унаследованных классах.

В этом случае возникают виртуальные методы.

Наследование: примеры

Разбираем второй пример кода к данной лекции

- 02_figures.cpp

Модификаторы доступа

public, protected, private

При описании полей своего класса это про доступ к лично вашим полям:

- `public` – «это видно всем»
- `protected` – «это видно мне и унаследованным от меня классам»
- `private` – «это видно только мне и больше никому»

public, protected, private

При наследовании это про то, как обойтись с наследством:

- public – оставить как было у родителя
- protected – ограничить public до protected
- private – ограничить всё до private

public, protected, private: примеры


Разбираем третий пример кода к данной лекции

- 03_public_protected_private.cpp

public, protected, private: заметки

- «Паблик Морозов» – класс, который через public-методы даёт доступ к protected-полям родительского класса.
- Если предполагается, что от класса будет иерархия наследования, то нужно аккуратно думать насчёт protected и private. Если всё объявить private, то наследники будут сильно ограничены в своих возможностях – можно ненароком разрушить всю идею наследования. Но protected поля наследники смогут менять в ходе своей работы так, как им захочется – можно ненароком разрушить всю идею инкапсуляции.

- friend – исключение из правил, имеет доступ ко всему, включая private-поля.
- С одной точки зрения – нарушение всей строгой конструкции, вносит бардак.
- С другой точки зрения – даёт точечный доступ к полям в случаях, где иначе пришлось бы городить public для всех.

A photograph showing the lower half of a person standing on a rocky, mountainous terrain. The person is wearing dark trousers and brown hiking boots with red laces. The ground is covered with rocks and some dry grass. The background is a clear, light blue sky.

“A friend is someone who can access
your private parts”

– C++

Разбираем последние примеры кода к данной лекции

- 04_access_violation.cpp
- 05_friend_access.cpp

Что стоит запомнить из лекции

- Наследование – это о том, как не писать повторно один и тот же код для «почти одинаковых» сущностей.
- Наследование – это о том, как переиспользовать готовые компоненты, доопределяя для них своё нужное поведение.
- При наследовании есть много аспектов, что и как наследовать – какую иерархию классов составить, что взять неизменное и что переопределить при наследовании, какие модификаторы доступа поставить на все поля и методы. Это уже не про технические средства, это про искусство проектирования системы.

<https://tinyurl.com/y6hpeo65>



ЧИПОЛЛИНО,
МАЛЬЧИК МОЙ!

ЛУК,
Я ТВОЙ ОТЕЦ!

