Наследование и полиморфизм

№ урока: 3 **Курс:** Java Essential

Средства обучения: Компьютер с установленной IntelliJ IDEA.

Обзор, цель и назначение урока

Рассмотрение и применение модификаторов доступа.
Рассмотрение понятия инкапсуляции и механизмов наследования.
Рассмотрение полиморфизма.

Изучив материал данного занятия, учащийся сможет:

- Понимать работу наследования.
- Применять различные модификаторы доступа.
- Отличать и применять основные формы полиморфизма

Содержание урока

1. Рассмотрение понятия наследования.
2. Обзор и применение модификаторов доступа.
3. Вызов конструктора базового класса.
4. Приведение к базовому типу.
5. Понятия UpCast и DownCast.
6. Рассмотрение понятия полиморфизма.
7. Использование герметизированных классов.

Резюме

- ООП – Объектно-ориентированное программирование – парадигма программирования, в которой основными концепциями являются понятия объектов и классов.
- Наследование – механизм объектно-ориентированного программирования (наряду с инкапсуляцией, полиморфизмом и абстракцией), позволяющий описать новый класс на основе уже существующего (родительского), при этом свойства и функциональность родительского класса заимствуются новым классом. При наследовании одного класса от другого, используется ключевое слово extends, а при наследовании класса от интерфейса – implements.
- Недостаток наследования – хрупкий базовый класс. Хрупкий базовый класс — фундаментальная проблема объектно-ориентированного программирования. Проблема хрупкого базового класса заключается в том, что малейшие правки в деталях реализации базового класса могут привнести ошибку в производные классы. В худшем случае это приводит к тому, что любая успешная модификация базового класса требует предварительного изучения всего дерева наследования, и зачастую невозможна (без создания ошибок) даже в этом случае.
- Рекомендуется использовать следующие пары:
  - ✓ Базовый класс – Производный класс
  - ✓ Супер класс – Подкласс или (сабкласс)
  - ✓ Родительский класс – Дочерний класс
  - ✓ Класс Родитель – Класс Потомок
- Класс может наследоваться только от одного класса (но можно делать многоуровневое наследование).
- Класс может наследовать не ограниченное количество интерфейсов.
- Модификаторы доступа – это ключевые слова, задающие объявленную доступность члена или типа. При помощи модификаторов доступа можно задать следующие пять уровней доступности:
  1) public – доступ к типу или члену возможен из любого другого кода в той же сборке или другой сборке, ссылающейся на него.

CyberBionic systematics
CyberBionic Systematics ® 2016
19 Mariny Raskovoy Str., 5 floor
Kyiv, Ukraine
t. +380 (44) 361-8937
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

2) protected – доступ к типу или элементу можно получить только из кода в том же классе или структуре, либо в производном классе.
3) private – доступ к типу или члену можно получить только из кода в том же классе или структуре.

- Полиморфизм – возможность объектов с одинаковой спецификацией иметь различную реализацию.
- Полиморфизм предоставляет подклассу способ определения собственной версии метода, определенного в его базовом классе, с использованием процесса, который называется переопределением метода (method overriding).
- Базовые классы могут определять и реализовывать методы, а производные классы могут переопределять их. Это означает, что они предоставляют свои собственные определение и реализацию.
- Если производный класс наследует от базового класса, то он приобретает все методы, поля, свойства и события базового класса. Проектировщик производного класса может выбирать из следующих возможностей:
  1) переопределить члены в базовом классе
  2) наследовать метод последнего базового класса без его переопределения
  3) определить новую реализацию этих членов, которая скрывает реализации базового класса.
- Если в производном классе метод переопределяется, то этот член вызывается даже в том случае, если доступ к экземпляру этого класса осуществляется как к экземпляру базового класса.
- Ключевое слово – final, которое предотвращает наследование. Если класс помечен как final (финальный), компилятор не позволяет наследовать от него. Считается, что класс герметизирован или «запечатан».

Закрепление материала

- Что такое наследование?
- Какие недостатки наследования вы знаете?
- Что такое модификаторы доступа и где их используют?
- Что такое ООП?
- Назовите основные парадигмы ООР.
- Что такое полиморфизм?
- Что такое Cast, UpCast, DownCast?
- Объясните назначение ключевого слова final?

Дополнительное задание

Задание
Используя IntelliJ IDEA, создайте проект.
Требуется:
Создайте класс Printer.
В теле класса создайте метод void print(String value), который выводит на экран значение аргумента.
Реализуйте возможность того, чтобы в случае наследования от данного класса других классов, и вызове соответствующего метода их экземпляра, строки, переданные в качестве аргументов методов, выводились разными цветами.
Обязательно используйте приведение типов.

Самостоятельная деятельность учащегося

Задание 1
Выучите основные конструкции и понятия, рассмотренные на уроке.

Задание 2
Используя IntelliJ IDEA, создайте проект.
Требуется:
Создать класс, представляющий учебный класс ClassRoom.

CyberBionic systematics
CyberBionic Systematics ® 2016
19 Mariny Raskovoy Str., 5 floor
Kyiv, Ukraine
t. +380 (44) 361-8937
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

Создайте класс ученик Pupil. В теле класса создайте методы void study(), void read(), void write(), void relax(). Создайте 3 производных класса ExcelentPupil, GoodPupil, BadPupil от класса базового класса Pupil и переопределите каждый из методов, в зависимости от успеваемости ученика.
Конструктор класса ClassRoom принимает аргументы типа Pupil, класс должен состоять из 4 учеников.
Предусмотрите возможность того, что пользователь может передать 2 или 3 аргумента.
Выведите информацию о том, как все ученики экземпляра класса ClassRoom умеют учиться, читать, писать, отдыхать.

Задание 3
Используя IntelliJ IDEA, создайте проект.
Требуется:
Создать класс Vehicle.
В теле класса создайте поля: координаты и параметры средств передвижения (цена, скорость, год выпуска).
Создайте 3 производных класса Plane, Car и Ship.
Для класса Plane должна быть определена высота и количество пассажиров.
Для класса Ship – количество пассажиров и порт приписки.
Написать программу, которая выводит на экран информацию о каждом средстве передвижения.

Задание 4
Используя IntelliJ IDEA, создайте проект.
Требуется:
Создайте класс DocumentWorker.
В теле класса создайте три метода openDocument(), editDocument(), saveDocument().
В тело каждого из методов добавьте вывод на экран соответствующих строк: "Документ открыт", "Редактирование документа доступно в версии Про", "Сохранение документа доступно в версии Про".
Создайте производный класс ProDocumentWorker.
Переопределите соответствующие методы, при переопределении методов выводите следующие строки: "Документ отредактирован", "Документ сохранен в старом формате, сохранение в остальных форматах доступно в версии Эксперт".
Создайте производный класс ExpertDocumentWorker от базового класса ProDocumentWorker. Переопределите соответствующий метод. При вызове данного метода необходимо выводить на экран "Документ сохранен в новом формате".
В теле метода main() реализуйте возможность приема от пользователя номера ключа доступа pro и exp. Если пользователь не вводит ключ, он может пользоваться только бесплатной версией (создается экземпляр базового класса), если пользователь ввел номера ключа доступа pro и exp, то должен создаться экземпляр соответствующей версии класса, приведенный к базовому – DocumentWorker.

Задание 5
Зайдите на сайт Oracle.
Используя поисковые механизмы Oracle, найдите самостоятельно описание темы по каждому примеру, который был рассмотрен на уроке, так, как это представлено ниже, в разделе «Рекомендуемые ресурсы», описания данного урока. Сохраните ссылки и дайте им короткое описание.

Рекомендуемые ресурсы

Oracle: Полиморфизм
https://docs.oracle.com/javase/tutorial/java/IandI/polymorphism.html

Oracle: Модификаторы доступа
https://docs.oracle.com/javase/tutorial/java/javaOO/variables.html

Oracle: Наследование
https://docs.oracle.com/javase/tutorial/java/IandI/subclasses.html

CyberBionic systematics
CyberBionic Systematics ® 2016
19 Mariny Raskovoy Str., 5 floor
Kyiv, Ukraine
t. +380 (44) 361-8937
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com