Лекция 2

Проблема множественного наследования решена в java путем убирания возможности множественного наследования

Extend – наследование в Java

Интерфейс

*Interface <Name> {*

*//метод*

*}*

Интерфейс содержит в себе описание функционала, который должен реализовать. Для этого используется ***implements***. Класс, который реализует интерфейс, должен содержать в себе все эти методы. Один класс может реализовать произвольное количество интерфейсов. К экземпляру класса можно обратиться по имени интерфейса, который этот класс реализует.

Практически все методы в java являются виртуальными, за исключением модификаторов static, private, final.

Изменяемые и не изменяемые классы

В java есть экземпляры, которые не могут быть изменены в процессе работы программы. Для создания неизменяемого класса необходимы требования:

* Модификатор final
* Все члены должны иметь модификатор final
* Ссылка this никуда не должна пропасть
* Все сложные объекты внутри класса (массивы, экземпляры и тд) должны быть инициализированы в момент создания класса и не должны никуда деваться.

Класс String является не изменяемым.

*String s = new String(“abc”);*

*a = a + “d”; //создается новый экземпляр класса (new String(“abcd”))*

*StringBuilder - изменяемый класс*

Указатели

В Java нет указателей.

Исключения

*Try {*

*}*

*Catch (<Exception class>) {*

*}*

*Finally {*

*}*

Для инициализации ошибки можно использовать throw <Exception class>.

Throws - используется при объявления метода. Необходимо для того, чтобы «пробросить» ошибки на более верхний уровень.

Ввод и вывод

Есть набор классов, которые позволяют считывать и выводить данные.

Для вывода есть класс PrintWriter. Умеет делать все

Для считывания есть класс Scanner. Позволяет считывать данные с любых потоков ввода. Сканер работает оооочень медленно. Для больших объемов он не рекомендуется к использованию.

Классы

В Java нет деструкторов

Класс создается при помощи new;

В Java используется сборщик мусора, который сам находит в памяти объекты, на которые никто не ссылается и удаляет его.

Сборщик мусора – процесс, которые чередует активный режим и неактивный. Существует возможность уничтожить объект. Класс может содержать в себе метод именуемый *finalize();*  Он вызывается, когда сборщик мусора удаляет объект.

Передача параметров в подпрограмму

В методы передается копия ссылки, т.е. все параметры передаются по значению, но можно изменить переменную, которая передавалась в качестве параметра.

Сериализация

Class A implements Serializable {

}

Это значит, что экземпляры класс можно вводить и выводить. Позволяет удобно сохранять объекты.

Хэшировавние

Идея хэширования состоит в упрощенном представлении объектов для возможности их сравнения в очень короткий срок.

Полиноминальное хеширование.

P > max(<состовляющие>) и является простым числом