**170620**

ООП программа - это совокупность объектов, которые взаимодействуют друг с другом путем передачи сообщений. У каждого объекта есть состояние и поведение. Состояние описывается набором полей у класса, а поведение методами класса.

Класс - синтаксическая единица, которая описывает структуру совокупности объектов, принадлежащих этому классу. Класс сам по себе тоже объект. Когда запускается программа создается объект класса. Запускаются объекты классов, а потом уже идет обращение к ним в случае необходимости для создания экземпляров класса. Загружается байт код создается объект класса, потом идет поиск точки входа main и он запускается.

Инкапсуляция объединение данных и кода в классе и закрытие доступа к ним с помощью модификаторов доступа. ООП позволил повысить уровень абстракции и позволил повторно использовать код за счет наследования и делегирования. Передача сообщений объекта другому объекту осуществляется благодаря наличию доступа к другому объекту т.е. ссылки на другой объект.

Полиморфизм: наследование, переопределение методов.

**Переменный в Java полиморфны: ссылочной переменной мы можем присвоить любой объект(экземпляр любого класса), лишь бы он обладал тем же типом что и переменная. То что хранится в переменной обладает определенным типом имеет соответствующий интерефейс и может отвечать на соответствующие запросы.**

ООП позволяет структурировать код для удобной работы с ним.

[Книги Тимоти Бадд] [Роберт С Мартин Чистый код] [Идеальный программист тот же автор]

Codingbat.com, HackerRank.com

[ДЗ] – 1 в файле.

160625[2]

Конструктор конструирует объекты, а не создает их. Их создает JVM. Каждый объект в джаве является монитором. У каждого объекта есть hashcod. При создании объекта отводится память потом идет обращение к конструктору суперкласса, потом срабатывает инициализатор родителя, потом конструктор родителя, потом инициализатор наследника и конструктор наследника. Инициализатор нужен, если у нас есть много разных конструкторов, но в них есть что-то общее. И это общее мы можем запихнуть в блок инициализации.

**Что отличает конструкторы от остальных методов? Отсутствие типа возвращаемого значения. Конструкторы не наследуются и не переопределяются.**

**[3]Конструкторы обладают всеми модификаторами доступа.**

Синглтон – это класс который гарантирует существование только одного своего экземпляра. Static переменные существуют в единственном числе.

Immutable объекты класса – это которые не изменяются (final) Можно безопасно использовать в потоках. При использовании final нет значения по умолчанию. Мы обязаны присвоить ему значение либо в конструкторе, либо в блоке инициализации. Поля инициализируются при создании классов значением по умолчанию. А локальные переменные нужно инициализировать.

Ссылочные типы – это адрес в динамической куче 4байта если 32 битная java машине, 64 в битной. 64 битные работают медленнее. Локальные переменные располагаются в стеке выполнения(массив из 32 или 64 бит ячеек). Туда помещаются переменные методов, которые выполняются в данный момент. Разница между ссылочными типами и примитивными типами состоит в том, что в примитивных хранится значение, а в адрес памяти в куче, где хранится объект.

**170627[06] ДЗ**

**170629 –** Синглтоный – Laizy initialization. В самом начале происходит инициализация полей класса и констант на этапе компиляции, потом статики. Объект(экземпляр) описывающий сам класс один, а потом создается множество экземпляров класса. Стек вызовов.

[03]-при компиляции вложенные кассы отделяются знаком доллара и именем класса. А анонимные классы просто номерами. Поэтому не рекомендуется использовать его в именах полей и т.д. Массив args существует он не будет null. Параметры в отличие от С++ не передаются. Методы экземпляра те где неявно присутствует ссылка на тот объект к которому обратились, и эта ссылка всегда называется this.

[4] ClassLoader – объет который загружает классы. С помощью наследования можно использовать повторный код. С помощью полиморфизма мы можем писать обобщенный код, который рабоет с произвольными объектами произвольных классов, главное, чтобы они обладали соответствующим типом. И за счет наследования мы можем менять реакцию(поведение) объектов на одни и те же сообщения. В каждом методе экземпляра класса неявно присутствует ссылка на тот объект, к которому идет обращение и эта ссылка называется this и она нам доступна.

Внутри классов могут быть: поля класса, поля экземпляра, методы класса, методы экземпляра, статический (класса) инициализатор, инициализатор экземпляра, вложенные(внутренние) классы, интерфейсы. В классе есть описание пакета, есть импорты для упрощения. Сокращения имен.

[05]

static void classMethod(){  
 synchronized (SyncExample.class){  
   
 }  
}  
 void instanceMethod(){  
 synchronized (this){  
  
 }  
}

Конструктор – это метод особого вида.

Class<A> c = A.class;

// A a = new A();  
  
 A a = c.newInstance();  
  
 Constructor<A> constructor = c.getConstructor(null);  
 constructor.newInstance(null);

// a.m(); эквивалент  
 m.invoke(a,null);

У каждой переменной есть время жизни Variable Scope. Hiding – некоторые сущности называются одинаково и закрывают некоторый объект соблой.