1. Цепочки вызовов

Ответ: пример в тетрадке

1. Класс path

Ответ: Объект пути содержит имена каталогов и файлов, которые составляют полный путь до файла/каталога, представленного объектом Path; Path содержит методы для извлечения элементов пути, манипуляций с ними и их добавления.

1. Чем отличается enum от класса констант

Ответ: Enum — это настоящий класс со всеми вытекающими из этого возможностями,но на Enum наложили одно серьезное ограничение — от него невозможно наследоваться. Кроме того, классы Enum безопасны по типу, тогда как статические поля - нет. Теперь возможна проверка времени компиляции по сравнению с потенциальными возможными ошибками во время выполнения с постоянным классом. Это также улучшает читаемость, потому что вместо ошибок, когда индекс списка констант недоступен

1. int a = true ? null : 1 (Почему скомпилится и чему будет равно а?)

Ответ: по дефолту переменной задается такое значение: int=0, Boolean=false, String=null, поэтому программа будет выводить 1.

1. принципы ООП: инкапсуляцию и наследование и полиморфизм и абстракция

Ответ:

1)Инкапсуляция (иногда называемая сокрытием информации) — это ключевое понятие для работы с объектами. Формально инкапсуляцией считается обычное объединение данных и операций над ними в одном пакете и сокрытие данных от других объектов.

2)Наследование - При использовании наследования вы говорите: Этот новый класс похож на тот старый класс. В коде это пишется как extends, после которого указываете имя базового класса. Тем самым вы получаете доступ ко всем полям и методам базового класса.

**3)Полиморфизм** – это способность программы идентично использовать объекты с одинаковым интерфейсом без информации о конкретном типе этого объекта.

4)Абстракция данных Абстра́кция в объектно-ориентированном программировании — это использование только тех характеристик объекта, которые с достаточной точностью представляют его в данной системе.

1. Что такое сокрытие данных ?

Ответ: это инкапсуляция

1. Что такое конструктор класса ?

Ответ:  специальный блок инструкций, вызываемый при создании объекта

1. SomeClass a = null  
   a.SomeClassMethod () // Сработает ? Обьяснить почему. Какой будет результат , если метод статичный

Ответ:

1. Как уронить класс с ошибкой стэковерфлоу( класс изначально пустой , конструктор по-умолчанию)  
   Class A { }

Ответ:

1. Дан класс  
   Class A {  
   MA()  
   MB()  
   MC()  
   }  
   Какие должны быть методы , чтобы монжо было сделать  
   [A.MA](https://vk.com/away.php?to=http%3A%2F%2FA.MA&cc_key=)().MB().MC();

Ответ:

1. Что такое new ?

Ответ: Оператор**new** создает экземпляр указанного класса и возвращает ссылку на вновь созданный объект.

1. ClassLoader ( типы и различия)

Ответ:Как известно, в Java реализована отложенная (или ленивая) загрузка классов. А это значит, что загрузка классов ссылочных полей загружаемого класса не будет выполняться до тех пор, пока в приложении не встретится явное к ним обращение. Иными словами, разрешение символьных ссылок не обязательно и по умолчанию не происходит. Тем не менее, в реализации JVM может использоваться и энергичная загрузка классов, т.е. все символьные ссылки должны быть учтены сразу. Вот для этого пункта и действует последнее требование.

**Типы загрузчиков Java**

В Java существует три стандартных загрузчика, каждый из которых осуществляет загрузку класса из определенного места:

1. **Bootstrap** – базовый загрузчик, также называется Primordial ClassLoader.

загружает стандартные классы JDK из архива rt.jar

1. **Extension ClassLoader** – загрузчик расширений.

загружает классы расширений, которые по умолчанию находятся в каталоге jre/lib/ext, но могут быть заданы системным свойством java.ext.dirs

1. **System ClassLoader** – системный загрузчик.

загружает классы приложения, определенные в переменной среды окружения CLASSPATH

1. Паттерны singleton, abstract fabric , fabric method

Ответ: Уверен, каждый программист слышал словосочетание "паттерны проектирования" (или шаблоны). Если очень коротко - это описание проблем, которые встречаются при написании объектно-ориентированного кода, а так же примеры решения этих проблем.

Паттерны создания объектов

Первая группа - это creational паттерны. Они в той или иной степени работают с механизмами создания объектов.

* [Singleton](http://www.javenue.info/post/2) - обеспечиваем существование в системе ровно одного экземпляра некоторого класса; ( Конструктор класса необходимо объявить с модификатором видимости private. Это предотвратит создание экземпляров класса как с помощью класса Singleton, так и с помощью его наследников. В связи с этим к объявлению класса смело можно дописать модификатор final. Паттерн Singleton полезен не только при работе с конфигурациями. Его можно использовать также при написании ConnectionPool, Factory и других вещей.)
* [Factory Method](http://www.javenue.info/post/17) - иными словами, Фабрика **делегирует** создание объектов наследникам родительского класса. Это позволяет использовать в коде программы не специфические классы, а манипулировать абстрактными объектами на более высоком уровне. (Используйте паттерн Factory в следующих случаях:

1)класс не имеет информации о том, какой тип объекта он должен создать;

2)класс передает ответственность по созданию объектов наследникам;

3)необходимо создать объект в зависимости от входящих данных;)

* [Prototype](http://www.javenue.info/post/70) - клонируем объекты на основании некоторого базового объекта;
* [Builder](http://www.javenue.info/post/58) - отделяем процесс создания комплексного объекта от его представления;
* [Abstract Factory](http://www.javenue.info/post/27) - Абстрактная фабрика предоставляет интерфейс для создания целых семейств объектов без указания конкретных классов. Объекты каждого семейства должны быть логически связаны между собой.( Используйте паттерн Абстрактная Фабрика в таких случаях:

1)система не должна зависеть от способа создания объектов;

2)система работает с одним из нескольких семейств объектов;

3)объекты внутри семейства взаимосвязаны.)

1. Class A{  
   private int i =0;  
   public A(){  
   i = 7  
   }  
   }  
   Последовательность присваивания значения i

Ответ:

1. Что такое абстрактный класс

Ответ:

Абстрактный класс создаются, как описание некоего общего состояния объекта.

1. У класса А есть метод dosmth(). Вопрос: можно ли его из нулл-экзкмпляра вызвать и если нет то как вызвать

Ответ:

1. Про вызов методов у null

Ответ: