1. Основные принципы ООП : инкапсуляция, наследование, полиморфизм
2. Инкапсуляция – получение или изменение данных путем вызова методов этого класса. Данные скрываются от прямого доступа с помощью модификаторов доступа.
3. Модификаторы доступа позволяют ограничить доступ к данным класса извне.
4. Private – члены класса доступны только членом данного класса;

Protected – члены класса доступны доступны всем классам одного пакета и наследникам другого пакета;

Friendly – члены класса доступны доступны всем классам одного пакета;

Public – члены класса доступны для всех классов в этом и других пакетах;

1. Наследование –возможность создания нового класса на основе текущего.
2. Полиморфизм – возможность использовать объекты с одинаковым интерфейсом без информации о типе и внутренней структуре объекта
3. Операторы циклов:

while

do … while

for

for each

1. Случайные числа генерируются с помощью метода Math.random().
2. Void methodName (тип … args) {} , где args– имя переменной.
3. Класс Object содержит следующие методы:

Protected Object clone()

Int hashCode()

String toString()

Protected void finalize()

Boolean equals()

getClass()

1. nt hashCode() –возвращает хэш-код объекта
2. String toString() – возвращает представление объекта в виде строки
3. Абстрактные классы содержат методы, которые не реализованы в этих классах, а будут реализованы в классах наследниках. Объявляются ключевым словом abstract.
4. Нельзя создать объект абстрактного класса. Абстрактные классы могут содержать абстрактные методы, которые будут реализованы в наследниках и полностью реализованные методы, а также конструкторы и поля данных.
5. Интерфейс – гарантирует , какие методы должен выполнять класс, имплементирующий данный интерфейс.
6. Интерфейс содержит по умолчанию поля final static, а все методы public.
7. Статические методы можно вызвать не создавая объект класса.
8. Статические переменные едины для всех объектов класса, при изменении в одном объекте, переменная меняется во всех.
9. Final класс нельзя наследовать, метод нельзя переопределить, переменную нельзя изменить.
10. Объект класса
11. Класс – описание совокупности объектов с общими переменными, методами и структурой написания.
12. Конструктор– это метод, который вызывается автоматически при создании объекта класса и выполняет действия только по инициализации объекта.
13. Для вызова тела одного конструктора из другого первым оператором вызывающего конструктора должен быть оператор this(переменные).
14. Конструктор имеет тоже имя, что и класс; вызывается только с ключевым словом new при создании объекта класса; конструктор не возвращает значение.
15. Перегрузка метода – методы имеют одинаковую сигнатуру, но разные входные данные.
16. Переопределение метода – методы имеют одинаковую сигнатуру но разную реализацию.
17. Память в java можно разделить на 2 части: Stack и Heap.

Stack хранит вызовы методов, Heap все остальное.

1. System.gc(); Runtime.getRuntime().gc(); finalize()
2. Структура кучи:

Eden – в этой области хранятся все создаваемые объекты.

Survivor space – сюда перемещаются объекты из предыдущего уровня, пройдя через первую чистку.

Tenured (old) generation – здесь скапливаются долгоживущие объекты.

Permanent generation – здесь хранится используемые классы, методы и т.д.

1. Git – облачное удаленное хранение.