**Сочетания клавиш в IDEA:**

**psvm** - быстрое создание метода public static void main(String[]args);

**sout** - быстрое создание System.out.print();

навести курсор на элемент строки и нажать ctrl+c - скопируется вся строка. То же самое с ctrl+x.

нажатие на колесико мышки позволяет удалять слова построчно.

**ctrl+y** - удаляет строку начиная снизу.

**ctrl+d** - дублирование клавиш.

**final** – аналог constant в C#. В таком случае переменная именуется с большой буквы, например, VARIABLE или MY\_VARIABLE.

**private** – члены класса доступны только внутри класса.

**default** – члены класса видны внутри пакета.

**protected** – члены класса доступны внутри пакета и в наследниках.

**public** – члены класса доступны всем.

**final** – запрещает другим классам наследоваться от текущего. В случае применения с методами ситуация аналогична.

**super** – аналог **base** в C#.

**this();** - ссылаемся на конструктор этого класса.

Для унаследования конструктора базового класса необходимо указать:

Upcast-Downcast выполняется следующим образом:

**package** inheritancesample;  
  
**public class** Program1 {  
 **public static void** main(String[] args) {  
 DerivedClass derived = **new** DerivedClass(5);  
 *// Upcast* BaseClass base = (BaseClass)derived;  
 *// Downcast* DerivedClass der = (DerivedClass)base;  
 }  
}

Для переопределения метода не нужно указывать ключевое слово **virtual**. Достаточно при определении указать аннотацию **@override**.

Цикл **foreach** выглядит следующим образом:

for(Element element : collection){}

Для реализации интерфейса нужно вместо ключевого слова **extends** указать **implements**.

Для операций с массивами используется класс **Arrays**.

Для копирования массивов в существующий массив используется метод **System.arraycopy(Object src, int srcPos, Object dest, int destPos, int length){}**

1 – исходный массив

2 – с какой позиции исходного массива начать копирование

3 – массив, в который копировать

4 – с какой позиции начать вставку элементов

5 – количество копируемых элементов

Для копирования массивов с возвратом нового массива используется **Array.copyOf(T[] original, int newLength){}**

1 – исходный массив

2 – размер возвращаемого массива

Для создания списков используется класс **ArrayList**.

Анонимные классы необходимы для множественного наследования.

BaseClass base = new BaseClass(){ new DerivedClass1{}, new DerivedClass2{}}

Значения параметра в ArrayList<?>, где ? может быть T, E, V, K.

T – type

E – element

V – value

K – key

**LinkedList<>** – применяется, когда нужно добавить элементы в начало коллекции.

**HashSet<>** - применяется, когда необходимо добавлять элементы с учетом сортировки и без повторений.

**TreeSet<>** - применяется, когда необходимо добавлять элементы с учетом сортировки по алфавиту.

**HashMap<Integer, String>** - ключ-значение