**Import …** - importing elements

**Import …\*** - importing all elements from the package

**Instanceof** – operator, which checks if this class extends another one

Static methods **static{…}** happens only once, when you create object

**....equals(…)** – идентичность по значению, == - по положению в памяти если это ссылка

**Scanner** scanner **= new Scanner(System.*in*);** - scanner variable (console in intelIdea)

Type … = scanner**.nextLine();** - similar to cin>> in c++

**Console** console **= System.console()** – variable for input in console (don’t know how to use)

Type … = console**.readLine()** – input in console

**<T, U, …>** - set method or class as multivariable (public class … <T> {…} , static <T> T …(){…} )

Class has to implement class “comparable” and override method compare to be able to compare

**<? Super/extends …>** - when you use collections in methods, you can add this to let pc understand that you will be using parents of child of this class

If you want to get inner class from class you have to write: **inner\_class … = new class().new inner\_class();**

**Strictfp class …** - class with only float values

**Abstract class …** - abstract class

**Final class …**  - final class

**void foo(char a, int... c)** – if you want to add unlimited value

в конструкторе можно вызвать другой конструктор класса методом **this(…);**

**enum имя {… , …}**  - объединения, могут иметь конструкторы и параметры **enum имя {…(…) , …(…); имя() {…}}**

**Color. …** - colors

**Data types and operations with them**

cover of types {

**Integer   
Byte   
Short  
Long   
Boolean   
Character   
Float   
Double**

}

**Integer.parseInt(…)** – string into the int

**Integer.valueOf(…)** – строку в число Integer

**new Intager(…) – parse int integer**

**try { }catch (){ }finally { }** – finally happens in both cases

**assert(…)** – short version of try catch, it checks if the equation(уравнение) is true and if not it breaks the program (you have to add **–ea** to the VM options for this thing to work)

**Catch has arguments:**

**NullPointerException** – when object is incorrectly used

**ArrayIndexOutOfBoundsException** – array exception

**ООП**

**Class … extends … {…}** – наследование

If parent’s constructor needs value you can enter it into the **super(…)**, with is placed in the start of the constructor of the extended class

You can assign extended class to the parent one

**Class …{ {…} – block …(){…} – constructor }** – block apply its code before the constructor

**DATE**

**Date** *date* **= new Date()** – creating data element

**Calendar** *calendar* **= Calendar.*getInstance*();** - creating variable*calendar***.setTime(***…***);** - setting time*calendar***.add(Calendar.*DAY\_OF\_MONTH*, …);** - add date to the current (1 element is defined variable of the Calendar and second is number of *calendar***.getTime();** - getting time

**SimpleDateFormat** *simpleDateFormat* **= new SimpleDateFormat("**dd/M/yy HH:m**");** - creating variable, which will be changing format of time (“dd/M/yy HH:m” – this abbreviator you can find by googling <https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html> )*simpleDateFormat***.format(***date***);** - changing date format

**Collections (interface)**

**ArrayList**

**List/LinkedList**

**Queue**

**Map/HashMap/LinkedHashMap/TreeMap**

**HashSet** – no repeated elements

Has methods:

**.add(…)** – add element

**.remove(…)** – remove element by id

**.get(…)** – get element by id

**.size()** - size

**PriorityQueue**

**.poll()** – get first element and delete it from array

**.peek()** – get first element

**Hashmap**

**Map … = new HashMap();** - creating variable

**.entrySet()** – swap into set massive

**.keySet()** – swap keys into Set massive

**ARRAYS**

**Int arr = new int […]** – creating array

**STRING**

**String … = new String(“…”);** - add a variable ( if you want to concat another str, it will create new variable in RAM) **don’t use if you are going to change it many times**

**StringBuilder … = new StringBuilder(“…”)** – add changeable variable( if you want to concat another str, it wont create new variable in RAM)

**Functions**

**System.gc();** - garbage collector, thing to clean useless rubbish but this function is dangerous to use

you can override method called “finalize”, which turns on, when garbage collector ends its work

**System.exit(0)** – end program

**Cloning**

If you want to clone this class you have to implement it from the **Cloneable** and override method:

**public class … implements Cloneable{  
 @Override  
 protected … clone() throws CloneNotSupportedException{ return (…) super.clone(); }**

**}**

**Create multivariable with many classes inside**

Create class (for example MyProxy), implement **InvocationHandler** and override **public Object invoke (Object proxy, Method method, Object[] args)** – cod there will happen before any classes methd

**InvocationHandler** handler **= new** MyProxy**();** - creating proxy

**Class[]** classes **= new Class[]{**Comparable**.class,** Callable**.class};** - creating array with classes **Object** proxy **= Proxy.*newProxyInstance*(null,** classes**,** handler**);** - creating variable, which includes those classes

**Сереализация – recording objects**

**Deadlock** – зависание программы, мол вызов одной функции, которая в ходе своей работы вызывает пред ведущую и замыкает цикл рекурсии