JAVAポイント解説 オラクルシルバーセミナー　Youtube　(https://youtu.be/-ih7h-x5bVI)

* + - * コマンドでの実行系は必須
      * Stringの参照の形について

String strA = ”Java”; はプールにオブジェクトを作るString strB=new String(”java”);はヒープにオブジェクトが生成されるので、strA == strBはfalse, ちなみにstrA.intern()==str Bはtrue

* + .intern()はプールになければ（あればその値）参照先をプールにコピーするメソッド

オブジェクトの個数を答える問題はガベージコレクションの対象になるオブジェクトはいくつあるか？とString参照（上記）だけしか出ない

* String Builderクラス（java.lang.StringBuilder---文字列の可変シーケンス）
* append(引数)；引数をStringBuilderの文字列シーケンスに追加
* insert(int offset、引数)；StringBuilderのoffset番目の位置に引数を挿入
* replace(int start、int end、String str)；start番目からend番目を文字列strで置換

↑Stringクラスのreplaceとの違いを確認しておく

ちなみにstart,offset,endの値は０番目からスタートする

* + - * 実行系トピック

Java HelloWorld.javaで実行可能（コンパイルなし）インメモリーで実行しクラスファイルは作られないことに注意！・必要条件としては１つのクラスであること、簡単なサンプルである事、同名のの.classファイルがない事

* + - * 変数の型とスコープ

プリミティブ型と参照型（初期値はnull）ちなみにnullはリテラル

インスタンス変数とローカル変数の違いを理解しておく、基本的には近い変数を見に行くが遠いインスタンスが見たい時などはthis.を使う、ちなみにstaticメソッド内ではthisは使えない

ゆるい型Local Variable Type interface

var i = 10;//varは型と推論してくれる。注意点として・ローカル変数のみで使える　・右辺の方が推論できない物は使えない（null、配列の初期化値）

* リテラル
* バイナリリテラル\_\_\_\_整数型（byte,short,int,long）を２進数で表記

Int i = 0b1010; またはint i2 = 0B1010;(2進数の接頭辞)、0xは１６進数、は0は８進数の接頭辞になる

* アンダースコア付きリテラル

数字を読みやすくするためだけのの区切り、コンパイル時に外されるもの

Int i = 0x5\_2;//16進数の52、float pi = 3.14\_15f;//フロート型の3.1415

* switchによる分岐処理
* 判定条件にStringを使用可能。switch文の文字列の比較にはequalsメソッドが使われてる
* switch文にはcontinueは使えない
* break、 defaultの使い方（breakなしだとcaseが処理される、default はcase以外の処理）

ちなみに、case ‘Y’ | ’X’ : これの意味は論理和（ビット演算子）でchar 123（{）を表している

* if文の三項演算子が読めるかどうか問われることがある（ネスト構造の場合あり）

配列

配列の宣言、生成、初期化子の仕方　（配列のインデックスは０から始まる）

コマンドライン引数mainメソッドの引数がString型で配列に入ること

多次元配列の初期化子を確認しておく

Java.util.ArrayListの宣言方法やメソッド確認いておく　　重要---要素の重複○

Java.util.HashMapの宣言方法やメソッド確認しておく　　重要---要素の重複不可（上書きになる）

ループ文

拡張for文は文法と使用目的（配列の要素を１つずつ取り出す）を押さえておく

インクリメント、デクリメントの前置と後置を理解しておく

static変数とstaticメソッド

カプセル化---概念を理解でおk

アクセス修飾子

オーバーロード　オートボクシング（変数の型の自動変換）

継承とインターフェイス

インターフェース

抽象クラス

参照型のキャスト

例外処理の基本、拡張 try-catchとthrows

Java.util.Listのデフォルトメソッド

ラムダ式の書き方（捨てるのもあり）

モジュールシステム---何のため？publicを制限するもの