1. SDK- SOFTWARE DEVELOPMENT KIT = jre + dev tools

JRE- JAVA RUNTIME ENVIREMENT = jvm + liby I inne pliki

JVM- JAVA VIRTUAL MACHINE: przyjmuje Java bytecode (.class) z libow JRE I uruchamia je ( tlumaczy na jezyk zrozumialy dla kompa) + garbage collector

Po uzyciu w kodzie slowa **import** classloader spradza czy klasa załadowana do JVM a jesli nie to laduje ja.

1. Plik .java jest kompilowany do pliku .class a class ladowany do JVM przez classloader
2. Jezyk obiektowy to:

-Abstrakcja

-Hermetyzacja

-Dziedziczenie

-Polimorfizm

1. Plik .java moze zawierac 1 klase publiczna. Nazwa pliku to nazwa tej klasy.
2. Uruchamianie: java + nazwa klasy z main.

Kompilacja javac + nazwa pliku .java

1. Niejawna konwercja: byte->short->int->long
2. + oraz += to jedyne przeciazone operatory w javie (dla Stringa)
3. Slowo kluczowe **instanceof** sprawdza czy obiekt jest instancja danej klasy/interface’u
4. Operator warunkowy/potrojny : int a = b<c?1:2 (jesli b<c to a=1 a jesli nie to a=2)
5. Return- konczy metode, break- konczy petle, continue- przechodzi do kolejnej iteracji
6. Metody klasy object:
   * clone
   * equals
   * hashCode
   * finalize
   * toString
   * getClass
   * notify
   * notifyAll
   * wait
7. modyfikatory dostepu:

* default: pakiet
* public: dostap z kazdego pakietu
* protected: pakiet I klasy pochodne
* private: tylko w klasie

1. STATIC: -dla pola

* dostep z kazdej instancji kalsy
* dostep przez nazwe klasy ( bez instancji)

-dla metody

* nie dziala na instancji + tez przez nazwe
* nie da sie jej nadpisac
* -blok static: moze byc wykorzystany do inicjalizacji pol klasy statycznej

1. FINALLY:

* Z klasy nie mozna dziedziczyc
* Metody nie mozna nadpisac
* Pola nie mozna zmienic

1. Sygnatura = nazwa metody + tym + argumenty (wazne przy przy nadpisywaniu/przeciazaniu)
2. Wyjatki : error I runtime – nie da sie ich obsluzyc. Exception(wyjatek kontrolowany): try/catch
3. THREAD: tworzy sie na 2 sposoby:

* dziedziczenie po Thread
* implementacja Runnable + stworzenie watku (new Thread() )
* w metodzie run() opisujemy dzialanie, metoda start() strartujemy
* synchronized

1. THREAD SAFE: jeżeli funkcje wywoływane jednocześnie przez wiele wątków wykonują się prawidłowo.
2. STRING:

* kazda klasa posiada metode toString()
* niemodyfikowalny (unmutable)
* thread safe ()

1. STRINGBUILDER

* Mutable
* Nie thread safe
* Wydajniejszy niz StringBuffer

1. STRINGBUFFER

* Mutable
* Thread safe

1. Autoboxing : konwersja int-> Integer

Unboxing : Integer->int

1. Pakiet java.io

* FileInputStream- czytanie binarne z pliku
* InputStreamReader- czytanie tekstowe z pliku
* BufferedReader, FileReader- czytanie liniami

ALTERNATYWNIE Z PISANIEM

1. COMPARABLE(sortowanie)

* Interface comparable implementuje metode compareTo() ktora zwraca :
  + - -1 jesli **this** < argument
    - 0 jesli rowne
    - 1 jesli >
* Sortuje sie za pomoca np Collections.sort()

1. SOLID