|  |  |
| --- | --- |
|  |  |
|  |  |
|  | Local variable, parameter, Instance variable, static variable  This – при использовании с переменными и методами указывает на текущий экземпляр (объект класса).  Нельзя использовать для обращения к переменным и методам внутри static метода или при присвоении значения static переменной. This – есть только у объектов.  Не является ошибочным обращение с помощью this к static элементам.  Garbage Collector - удаляет из памяти все неиспользуемые данные, ссылки на которые отсутствуют или не используются. Удаляет по своей логике, на которую повлиять нельзя. |
| **Lesson 13** | |
| **SWITCH** | Switch (expression) {  Case(): action  Break  Case(): action  Break  ….  Default: ….  }  Switch - может работать только со следующими типами данных: ***byte, short, int, char, string.*** Могут использоваться выражения (5\*10, 2+string). Если используются переменные, то они должны быть константами, т.е. определены с помощью final и присвоение должно быть в момент объявления. |
| **Lesson 14** | |
| **For**  **Foreach** | For (int i=1; i<=10; i++) { action }  for (int i=1, j=2, k=10; i <= 10 && i>-3 || i>4; i+=2, j+=4) – можно использовать несколько переменных/условий(не должно быть противоречий, и должно быть достижимым)/изменений  !-переменные должны быть одного типа  Break – для выхода из цикла  Continue – продолжает цикл не включая текущую итерацию  OUTER: for (int hour=0; hour<=23 ;hour++) – цикл можно называть, потом можно обращаться и использовать в цикле ({break OUTER;}) |
| **Lesson 15** | |
| **While**  **Do while** | ***While(i<=10) {…};***  Если в While(i++ <= 10) {…} используем увеличение переменной, то действие происходит даже если в этой итерации вернуло false, т.е. i=11.  ***Do{ action } while(i<=10);***  - отличается от while тем, что действие по любому выполняется, т.е. одно выполнение будет всегда  !- можно использовать вложенные циклы различных типов, напр. Do{ for(…) {…};} while(…); |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |