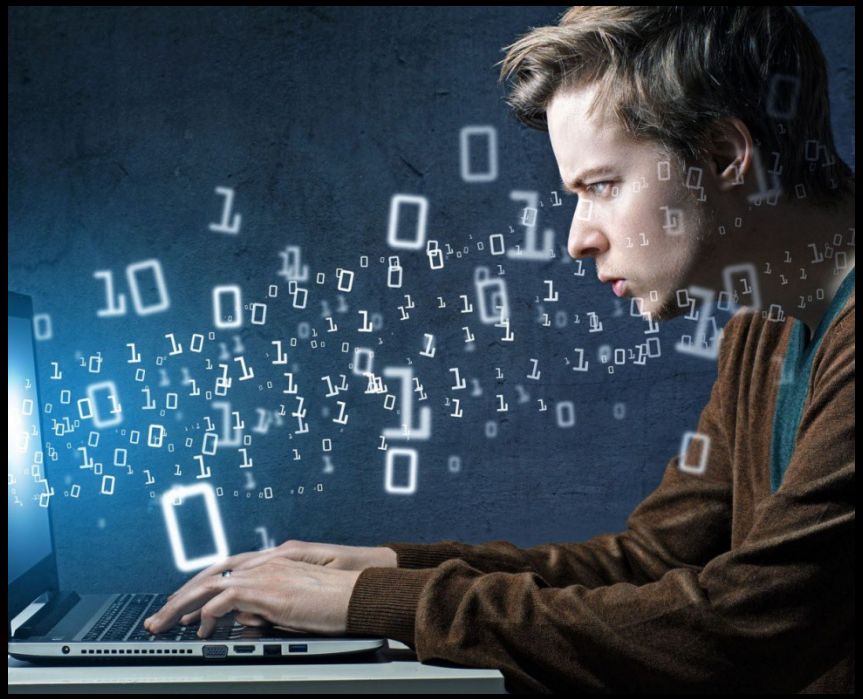
Soft Intellect

Основи на програмирането

Hello World…

Soft Intellect

Програмист ?!



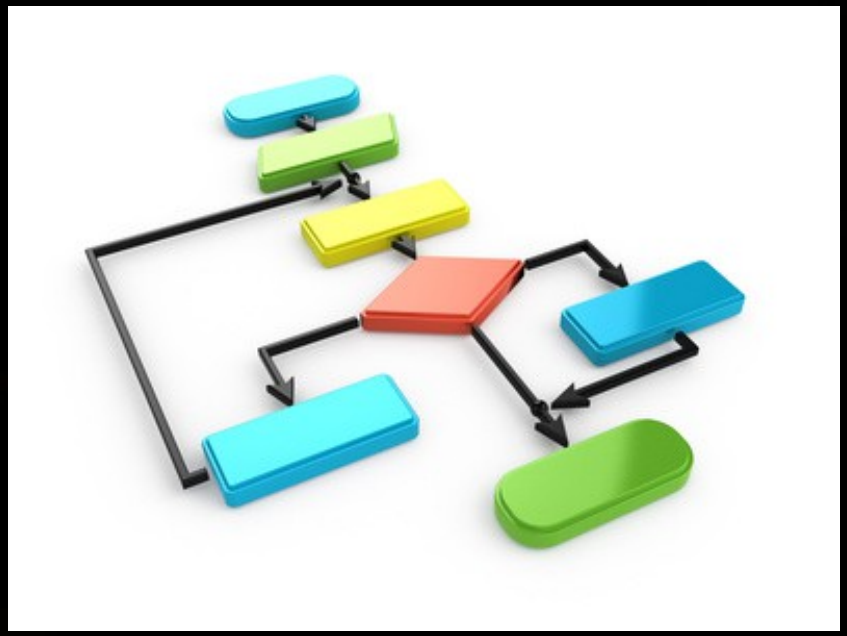
Soft Intellect

Алгоритми

- Какво означава алгоритъм
- Как се изгражда един алгоритъм
- Как се тества един алгоритъм

Soft Intellect

Демонстрация на алгоритъм



Soft Intellect

Платформата JAVA

- Платформа за изпълнение на JAVA код
- JVM (Java Virtual Machine) – Java виртуална машина
- BYTECODE

Soft Intellect

Първата ви JAVA програма

Soft Intellect

Променливи в JAVA

- Информацията, която се намира в паметта на компютъра, се съхранява в променливи
- Променливи имат **име**, **тип** и **стойност**
- Типовете променили в JAVA са **стойностни** и **референтни**
- Променливите съществуват в определен обхват

Soft Intellect

Променливи в JAVA

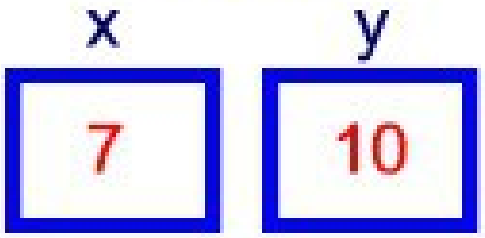
- Променливата е контейнер на информация, която може да сменя стойността си
- Именуване на променливите
- Деклариране на променливи
- Присвояване на променливи – извършва се с оператора „равно“ → =

Променливи в JAVA - DEMO

CODE:

```
int x = 7;
int y = 10;
```

MEMORY:



Soft intellect

Примитивни типове данни

- **byte** – стойностен тип 8 бита, default: 0, [-128, +127]
- **short** – стойностен тип 16 бита, default: 0, [-32 768, +32 767]
- **int** – стойностен тип 32 бита, default: 0, [-2 147 483 648, +2 147 483 647]
- **long** – стойностен тип 64 бита, default: 0L, [-9 223 372 036 854 775 808, +9 223 372 036 854 775 807]
- **float** – стойностен тип 32 бита, default: 0.0f, [-3.4E+38, +3.4E+38]
- **double** – стойностен тип 64 бита, default: 0.0d, [-1.7E+308, +1.7E+308]
- **boolean** – стойностен тип 1 бита, default: false, true/false
- **char** – стойностен тип 16 бита, default: ‘\u0000’, [0, +65 535]

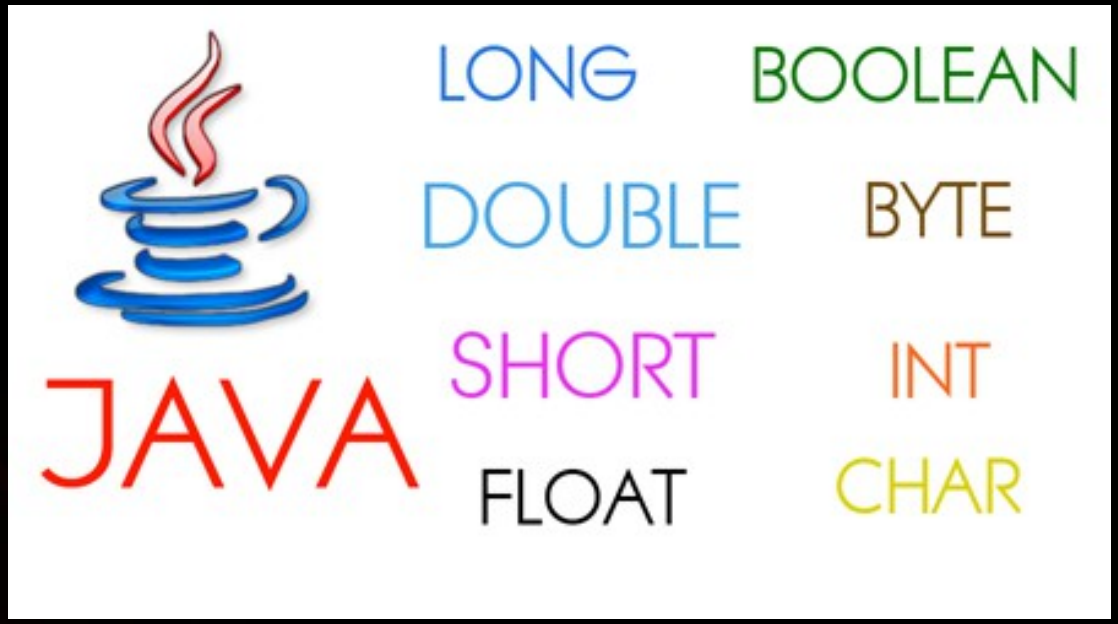
Soft Intellect

Видове примитивни типове

- Целочислени типове – **byte**, **short**, **int**, **long**
- Реални типове с плаваща запетая – **float** и **double**
- Булев тип – **boolean**
- Символен тип – **char**

Soft intellect

Примитивни типове - DEMO



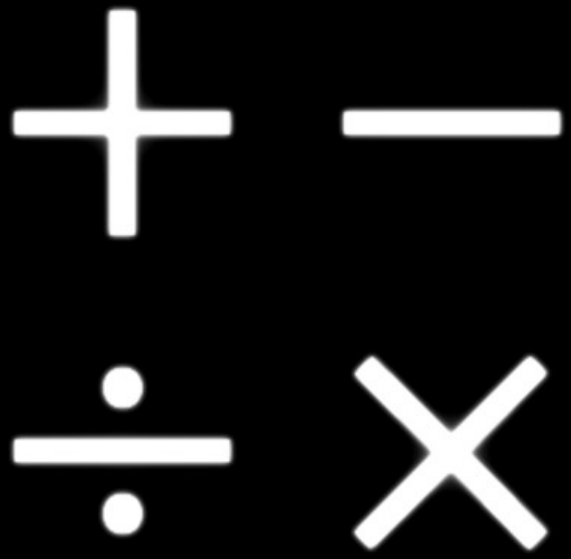
Soft Intellect

Аритметични оператори в JAVA

- Аритметичните оператори представят математически операции
- Стандартни математически оператори +, -, *, /
- Модулен оператор за остатък от делене %
- Съкратен запис за инкрементиране с 1 ++
- Съкратен запис за декрементиране с 1 --

Soft intellect

Аритметични оператори - DEMO



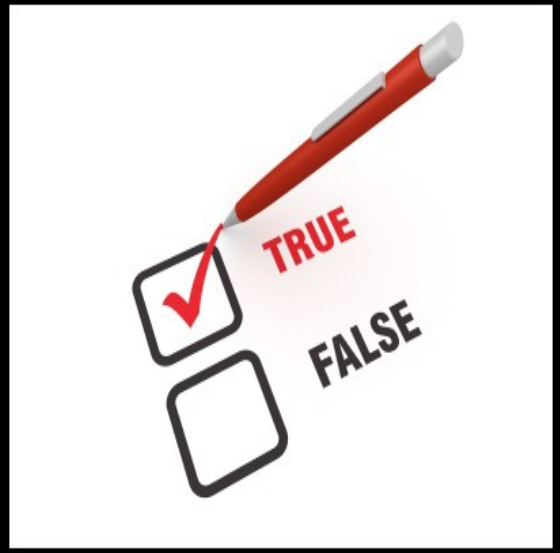
Soft Intellect

Сравнение в JAVA

- Сравнението в JAVA се извършва с операторите за сравнение:
  - ==
  - !=
  - <
  - >
  - <=
  - >=
- Резултатът от сравнението винаги е boolean

Soft intellect

Сравнение - DEMO



Soft Intellect

Логически оператори в JAVA

- Извършват се върху boolean променливи или изрази за сравнение (защото те връщат boolean)
  - && - логическо И
  - || - логическо ИЛИ
  - ! - оператор за отрицание
- Приоритет на операторите: !, ^, &&, ||

Soft Intellect

Логически оператори - DEMO

A	B	A||B	A&&B	A^B	!A
false	false	false	false	false	true
true	false	true	false	true	false
false	true	true	false	true	true
true	true	true	true	false	false

Soft Intellect

Конструкция IF..ELSE..

- if условието приема променлива от тип boolean или операция, която връща boolean (сравняване, логически операции)
- Ако условието върне положителен (true) резултат се изпълнява if блокът
- Ако условието върне негативен (false) резултат се изпълнява else блокът
- else блокът не е задължителен
- Възможно е проверяването на няколко отделни условия чрез конструкция
- if .. else if .. else if .. else

Soft Intellect

Конструкция IF

```
if (булев израз) {
    тяло на условната конструкция
}
```

- Значимост на къдравите скоби

Soft intellect

Конструкция IF..ELSE..

```
if (булев израз) {
    тяло на условната конструкция
} else {
    тяло на else-конструкцията
}
```

Soft Intellect

Конструкция IF..ELSE IF..ELSE

```
if (булев израз) {
    тяло на условната конструкция
} else if (булев израз) {
    тяло на else if-конструкцията
}
else {
    тяло на else-конструкцията
}
```

Soft Intellect

IF конструкции - DEMO

Soft Intellect

Цикли

- Циклите повтарят даден блок от код.
- Едно завъртане на цикъл се нарича итерация.
- Циклите се изпълняват, докато условието на цикъла е вярно.
- Помагат за пресмятане и изчерпване.
- Цикли, които се повтарят безкрайно пораждат грешка.

Soft Intellect

Цикълът FOR () {…}

- Цикълът for се състои от 4 части:

✓ Инициализация; //не е задължителна

✓ Проверка на условието //не е задължителна

✓ Помяна //не е задължителна

✓ Тяло на цикълът // задължителна

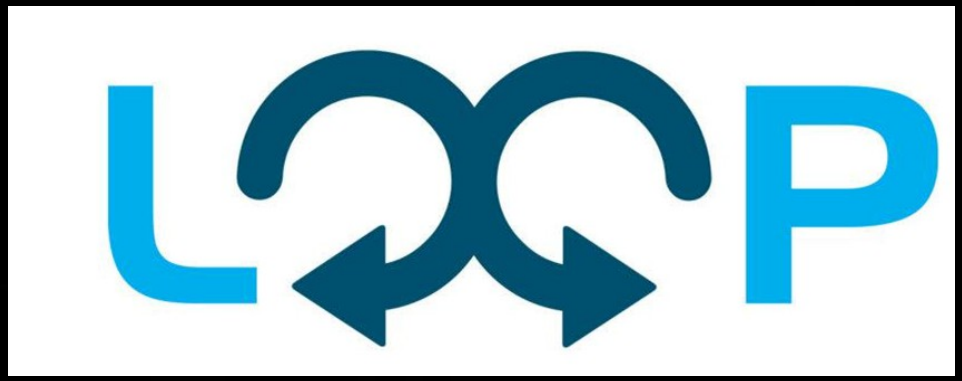
Soft Intellect

Цикълът FOR () {…}

```
for (initialization; check; incrementation)
{
    тяло на цикъла
}
```

Soft Intellect

Цикълът FOR () {…} – DEMO



Soft Intellect

Операторът BREAK

- Оператор BREAK прекъсва изпълнението независимо от условието на цикъла
- Важи за всички типове цикли
- Кодът в цикъла след break не се изпълнява
- Използвайте break само, когато се налага

Soft Intellect

Операторът BREAK - DEMO

Soft Intellect

Операторът CONTINUE

- Предизвиква нова итерация
- Кода под цикълът не се изпълнява
- За разлика от оператора break, при continue цикълът продължава да се изпълнява

Soft Intellect

Операторът CONTINUE - DEMO

Soft Intellect

Pair Programming or Code Review

- Pair Programming – един програмист пише код, докато друг го наблюдава и проверява
- Code Review – проверка на кода, който сте написали от ваш колега

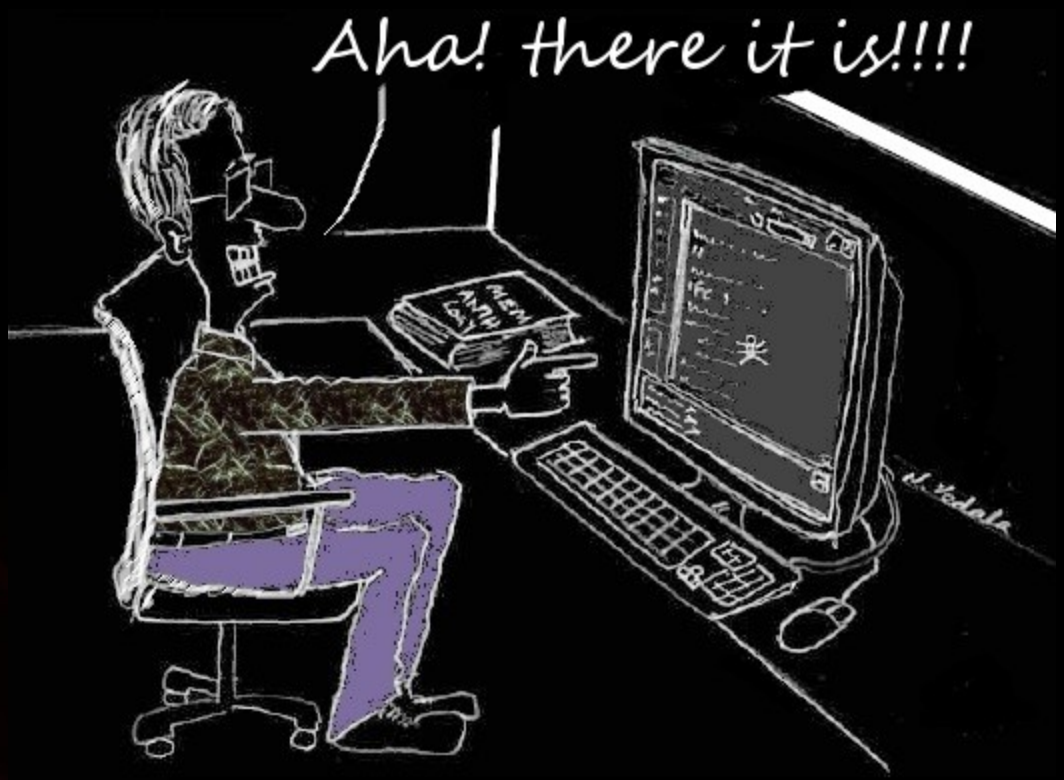
Soft Intellect

Тестване и Debugging

- Винаги проверявайте сами кода, който се написали
- Разпишете си различни от стандартните test cases и пробвайте с тях
- Пробвайте с крайни и невалидно стойности
- При проблем – дебъгвайте, по-ефективно е от гледането на кода

Soft Intellect

Тестване и Debugging - DEMO



Soft Intellect

Unit Testing

- Unit тестовете представляват код който тества друг код
- Има две основни функции:
  - Лесно откриване на проблем, след промяна на кода
  - Запазване на състоянието на работеща система, като се покриват основни функционалности

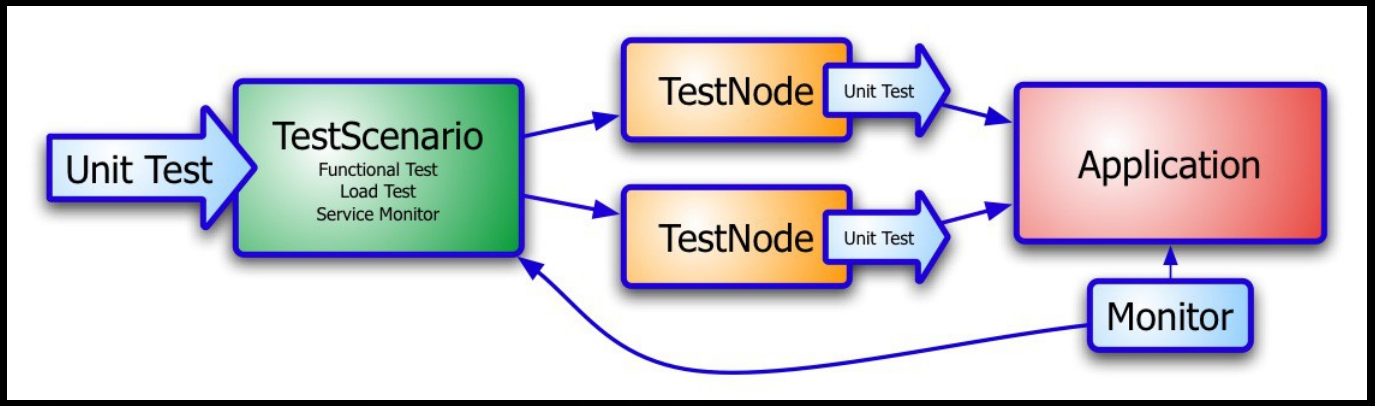
Soft Intellect

Unit Testing

- Позволява лесно да се мине през целия код при различни сценарии
- Лесна поддръжка на кода
- Драстично намаляване на бъговете
- Лесно се разбира за какво се използва даден код
- Unit тестовете се пишат от ПРОГРАМИСТИ!

Soft Intellect

Unit Testing - Демо



Soft Intellect

Test Driven Development

- Първо се пишат Unit тестове
- После се пише програмния код на приложението
- Тестовете се пускат, за да проверят до колко написания код е верен
- Тестовете също търпят някакви промени

Soft Intellect

Въпроси