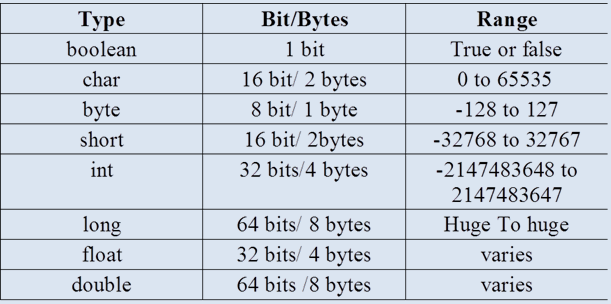
**Documentatia limbajului SLS**

Acest ghid are ca scop prezentarea conceptelor si sintaxei limbajului SLS (Simplitate la Superlativ), cuprinzand toate informatiile necesare pentru a rula in cele mai bune conditii aplicatii bazate pe acesta. Limbajul SCS este o reinterpretare a Java, mostenind astfel caracteristicile sale: este un limbaj object-oriented, avand ca scop simplificarea conceptului de programare nativa pe android si nu numai, oferind o buna rampa de lansare pentru incepatori.

**Sectiunea 1.1: *Tipuri de variabile***



**Sectiunea 1.2: *Atribuirea unei valori pentru o variabila existenta***

|  |  |
| --- | --- |
| **Atribuirea unei valori** | a = 7; |

**Sectiunea 2.0: *Structuri de date si exemple***

Avand ca scop introducerea in lumea programarii si creerea unui mediu facil de a invata, SLS foloseste structuri de date familiare, incercand astfel sa devina aproape la fel de intuitiva ca un limbaj pseudocod. Pentru insipiratie, am folosit atat Java cat si C++ si Python.

**Sectiunea 2.1: *Structura de decizie ( if )***

**if** (a > 4) :

a = 10;

b = 7;

**else**

**if** (a == 4) :

a = 3;

b = 2;

**else** :

a = 1;

b = -2;

**end;**

**Sectiunea 2.2: *Structura “cat timp” (while)***

**while (a > 5) :**

**// cod**

**end;**

**Sectiunea 2.3: *Structura “pentru” (for)***

**for ( i = 0; i < 10; i++) :**

**// cod**

**end;**

**Sectiunea 3.0: *Definirea functiilor si claselor noi***

Fiind un limbaj orientat pe obiecte si clase, SLS le permite utilizatorilor sa isi creeze singuri functii si obiecte noi. Sintaxa obiectelor este intuitiva, fiind un hibrid intre Java si C++, pastrand totusi conceptul de baza, simplitatea.

**Sectiunea 3.1: *Initializarea unei functii***

Definirea unei functii

// functia se incheie la apelarea “return”

Function int adunare (int a, int b) :

return a + b;

Apelarea unei functi

Int c = adunare(x, y);

**Sectiunea 3.2: *Initializarea claselor si obiectelor***

Definirea obiectului “car”

new Object car :

variables :

engineSize : 2000

year : 2014

seats : 2

methods :

calculateAge

Initializarea unul obiect de tip car

Car myCar = new Car();

Se genereaza automat functii de tip getter and setter

myCar.setEngineSize(1000);

print(myCar.getEngineSize());

**Sectiunea 4.0:Biblioteca standard de functii predefinite**

void print(String str) - Printeaza str in consola

String getStringInput() - citeste un string de la tastatura

int getIntInput() - citeste un int de la tastatura

float getFloatInput() - citeste un float de la tastatura

boolean getBooleanInput() - citeste un boolean de la tastatura

boolean prim (int x) - Returneaza daca variabila x este numar prim

int cmmdc(int x,int y) - Returneaza cel mai mare divizor comun dintre x si y

int cmmmc(int x, int y) - Returneaza cel mai mic multiplu comun dintre x si y

double putere (int x, double y) - Returneaza y la puterea x

int factorial(int x) - Returneaza x factorial

readFile(String filename) - Returneaza un String care este format din continutul fisierului de la path-ul filename

writeToFile(String filename, String input) - Adauga in fisierul de la path-ul filename String-ul input

**Sectiunea 5.0 : Citire si scriere pentru fisiere**

Cu ajutorul functiilor predefinite.