

Boa - Beginner

- Level 1 -

Basic Types in Boa

- ***bool*** - Typical Boolean as you'd find in other languages, evaluates to true or false (ex; true)
- ***string*** - Array of Unicode characters, (ex; "hello")
- ***int*** - 64-bit signed integer
- ***float*** - 64-bit IEEE floating point value
(notice the larger quantity of bits allocated to int and float values)
- ***time*** - unsigned integer representing the number of microseconds since 00:00:00 UTC 1 January 1970
- In a future lecture the Data Structure Types will be covered...

Domain-Specific Types

- Project
 - id : string
 - name : string
 - programming_languages : array of string
 - code_repositories : array of CodeRepository
- Person
 - email : string
 - real_name : string
 - username : string
- CodeRepository, Revision, etc.
- Generic enough to support any object-oriented language

Output Aggregators

All of these aggregators end in “of T” this denotes the type of the thing being aggregated. Indices can be used to reference elements of the projects to do more specific operations on the project data.

From the previous lecture:

Sum - sum [indices] of T Computes the sum of the data.

Top - top (param) [indices] of T [weight] Records the top (param) number of elements.

New:

Bottom - bottom [param] [indices] of T [weight] - Gets the bottom (param) elements

Collection - collection [indices] of T - Collects data points without performing any operations.

Set - set (param) [indices] of T - A set of data. Param denotes an optional specified limit to the number of results. The set contains the first *param* elements if specified.

Mean - mean [indices] of T - Calculates average of the data.

Minimum - minimum [param] [indices] of T [weight] - gets the lowest (param) sets of weighted elements

Maximum - maximum [param] [indices] of T [weight] - gets the highest (param) sets of weighted elements

Quantifiers

- Loop with conditions
- General form:

```
1  quantifier (var: type; condition)
2      statement
```

- If there exists a value of i where condition holds, then run body once (with i bound to one matching value)
- Condition must be a ***bool*** statement
- 3 types: foreach, exists, ifall

One Example!

- What are the five most used licences?
 1. Licences → array of string type
 - a. licenses is an attribute of Project
 - b. Must iterate through array using **quantifier**
 2. Five most → **top** aggregator
 3. Most used → need to **count** # licences

One Example!

```
1 # Counting the 5 most frequently used licenses
2 p: Project = input;
3 counts: output top(5) of string weight int;
4
5 foreach (i: int; def(p.licenses[i]))
6     counts << p.licenses[i] weight 1;
```

Resources

- Microsoft Presentation: https://youtu.be/KZi_KdWv8ns