## Module sui::object bag

Similar to sui::bag, an ObjectBag is a heterogeneous map-like collection. But unlike sui::bag, the values bound to these dynamic fields must be objects themselves. This allows for the objects to still exist in storage, which may be important for external tools. The difference is otherwise not observable from within Move.

Creates a new, empty bag

Adds a key-value pair to the bag bag: & mut ObjectBag Aborts with sui::dynamic field::EFieldAlreadyExists if the bag already has an entry with that key k: K.

Immutably borrows the value associated with the key in the bag bag: & ObjectBag. Aborts with sui::dynamic field::EFieldDoesNotExist if the bag does not have an entry with that key k: K. Aborts with sui::dynamic field::EFieldTypeMismatch if the bag has an entry for the key, but the value does not have the specified type.

Mutably borrows the value associated with the key in the bag bag: & mut ObjectBag. Aborts with sui::dynamic field::EFieldDoesNotExist if the bag does not have an entry with that key k: K. Aborts with sui::dynamic field::EFieldTypeMismatch if the bag has an entry for the key, but the value does not have the specified type.

Mutably borrows the key-value pair in the bag bag: & mut ObjectBag and returns the value. Aborts with sui::dynamic field::EFieldDoesNotExist if the bag does not have an entry with that key k: K. Aborts with sui::dynamic field::EFieldTypeMismatch if the bag has an entry for the key, but the value does not have the specified type.

Returns true iff there is an value associated with the key k: K in the bag bag: & ObjectBag

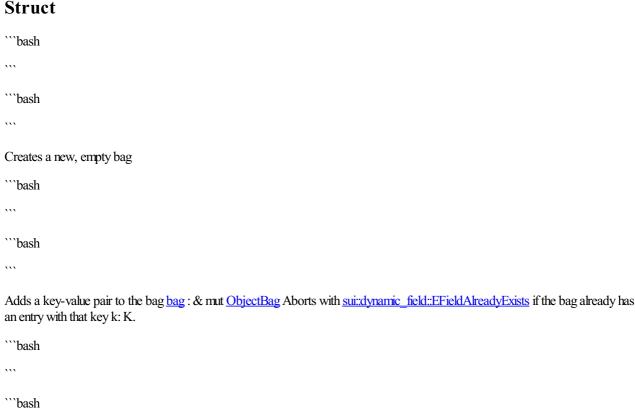
Returns true iff there is an value associated with the key k: K in the bag bag: & ObjectBag with an assigned value of type V

Returns the size of the bag, the number of key-value pairs

Returns true iff the bag is empty (if <u>length</u> returns 0)

Destroys an empty bag Aborts with **EBagNotEmpty** if the bag still contains values

Returns the ID of the object associated with the key if the bag has an entry with key k: K Returns none otherwise



...

<u>sui::dynamic_field::EFieldDoesNotExist</u> if the bag does not have an entry with that key k: K. Aborts with <u>sui::dynamic_field::EFieldTypeMismatch</u> if the bag has an entry for the key, but the value does not have the specified type.
```bash
```bash
···
Mutably borrows the value associated with the key in the bag bag: & mut ObjectBag. Aborts with sui::dynamic_field::EFieldDoesNotExist if the bag does not have an entry with that key k: K. Aborts with sui::dynamic_field::EFieldTypeMismatch if the bag has an entry for the key, but the value does not have the specified type.
```bash
```bash
Mutably borrows the key-value pair in the bag <u>bag</u> : & mut <u>ObjectBag</u> and returns the value. Aborts with <u>sui::dynamic_field::EFieldDoesNotExist</u> if the bag does not have an entry with that key k: K. Aborts with <u>sui::dynamic_field::EFieldTypeMismatch</u> if the bag has an entry for the key, but the value does not have the specified type.
```bash
```bash
Returns true iff there is an value associated with the key k: K in the bag $\underline{bag}$ : & $\underline{ObjectBag}$
```bash
```bash
Returns true iff there is an value associated with the key k: K in the bag bag: & ObjectBag with an assigned value of type V
```bash
```bash
Returns the size of the bag, the number of key-value pairs
"bash
***
```bash
V4311

Immutably borrows the value associated with the key in the bag  $\underline{bag}$ : &  $\underline{ObjectBag}$ . Aborts with

Returns true iff the bag is empty (if <u>length</u> returns 0)
```bash
```bash
Destroys an empty bag Aborts with <u>EBagNotEmpty</u> if the bag still contains values
```bash
```bash
Returns the ID of the object associated with the key if the bag has an entry with key k: K Returns none otherwise
```bash
```bash
Constants
```bash
Creates a new, empty bag
```bash
```bash
Adds a key-value pair to the bag $\underline{bag}$ : & mut $\underline{ObjectBag}$ Aborts with $\underline{sui::dynamic\_field::EFieldAlreadyExists}$ if the bag already has an entry with that key k: K.
```bash
···
```bash
Immutably borrows the value associated with the key in the bag bag: & ObjectBag. Aborts with <a href="mailto:sui::dynamic_field::EFieldDoesNotExist">sui::dynamic_field::EFieldDoesNotExist</a> if the bag does not have an entry with that key k: K. Aborts with <a href="mailto:sui::dynamic_field::EFieldTypeMismatch">sui::dynamic_field::EFieldTypeMismatch</a> if the bag has an entry for the key, but the value does not have the specified type.
```bash
```bash

sui::dynamic field::EFieldDoesNotExist if the bag does not have an entry with that key k: K. Aborts with sui::dynamic field::EFieldTypeMismatch if the bag has an entry for the key, but the value does not have the specified type. ```bash ```bash Mutably borrows the key-value pair in the bag bag: & mut ObjectBag and returns the value. Aborts with sui::dynamic field::EFieldDoesNotExist if the bag does not have an entry with that key k: K. Aborts with sui::dynamic field::EFieldTypeMismatch if the bag has an entry for the key, but the value does not have the specified type. ```bash ```bash Returns true iff there is an value associated with the key k: K in the bag bag: & ObjectBag ```bash ```bash Returns true iff there is an value associated with the key k: K in the bag bag: & ObjectBag with an assigned value of type V ```bash ```bash Returns the size of the bag, the number of key-value pairs ```bash ```bash Returns true iff the bag is empty (if length returns 0) ```bash ,,, ```bash

Destroys an empty bag Aborts with <a href="mailto:EBagNotEmpty">EBagNotEmpty</a> if the bag still contains values

Mutably borrows the value associated with the key in the bag bag: & mut ObjectBag. Aborts with

Mutably borrows the key-value pair in the bag  $\underline{bag}$  : & mut  $\underline{ObjectBag}$  and returns the value. Aborts with

sui::dynamic_field::EFieldDoesNotExist if the bag does not have an entry with that key k: K. Aborts with sui::dynamic_field::EFieldTypeMismatch if the bag has an entry for the key, but the value does not have the specified type.
```bash
···
```bash
Returns true iff there is an value associated with the key k: K in the bag $\underline{bag}$ : & $\underline{ObjectBag}$
```bash
```bash
Returns true iff there is an value associated with the key k: $K$ in the bag $\underline{bag}$ : & $\underline{ObjectBag}$ with an assigned value of type $V$
```bash
```bash
Returns the size of the bag, the number of key-value pairs
```bash
```bash
Returns true iff the bag is empty (if <u>length</u> returns 0)
```bash
```bash
Destroys an empty bag Aborts with <u>EBagNotEmpty</u> if the bag still contains values
```bash
```bash
Returns the ID of the object associated with the key if the bag has an entry with key k: K Returns none otherwise
```bash

Function  Adds a key-valse pair to the bag bag: & mut ObjectBag Aborts with suirstynamic_field-EFieldAleadyExists if the bag already has an entry with that key k: K.  "bash  ""bash	```bash
Adds a key-value pair to the bag bag: & mot ObjectBag Aborts with suindynamic field-IF-ieldAlreadyLixiss if the bag already has an entry with that key k: K.  "bash  ""bash  "	
an entry with that key k: K.  "bash  ""bash  "	Function
"bash ""  Immutably borrows the value associated with the key in the bag bag: & ObjectBag. Aborts with suicdynamic_field:FFeklDoesNorExist if the bag does not have an entry with that key k: K. Aborts with suicdynamic_field:FFeklDoesNorExist if the bag has an entry for the key, but the value does not have the specified type. "bash ""  Mutably borrows the value associated with the key in the bag bag: & mut ObjectBag. Aborts with suicdynamic_field:FFeklDoesNorExist if the bag does not have an entry with that key k: K. Aborts with suicdynamic_field:FFeklDoesNorExist if the bag has an entry for the key, but the value does not have the specified type. "bash ""  "bash ""  Mutably borrows the key-value pair in the bag bag: & mut ObjectBag and returns the value. Aborts with suicdynamic_field:FFeklDoesNorExist if the bag does not have an entry with that key k: K. Aborts with suicdynamic_field:FFeklDoesNorExist if the bag does not have an entry with that key k: K. Aborts with suicdynamic_field:FFeklDoesNorExist if the bag has an entry for the key, but the value does not have the specified type. "bash ""  Bash ""	
"bash ""bash suicdynamic field-EFieldDoesNotExist if the bag does not have an entry with that key k: K. Aborts with suicdynamic field-EFieldDoesNotExist if the bag does not have an entry with that key k: K. Aborts with suicdynamic field-EFieldDoesNotExist if the bag has an entry for the key, but the value does not have the specified type. "bash ""bash ""Bas	```bash
Immatably borrows the value associated with the key in the bag bag: & ObjectBag. Aborts with sairdynamic field:PFieldDoesNotExist if the bag does not have an entry with that key k: K. Aborts with sairdynamic field:PFieldTypeMismatch if the bag has an entry for the key, but the value does not have the specified type. "bash ""  "bash ""  Mutably borrows the value associated with the key in the bag bag: & mut ObjectBag. Aborts with sairdynamic field:PFieldTypeMismatch if the bag has an entry for the key, but the value does not have the specified type. "bash ""  "	
Immatably borrows the value associated with the key in the bag bag: & ObjectBag. Aborts with suicdynamic field-EFieldTypeMismatch if the bag has an entry for the key, but the value does not have the specified type. "bash ""bash ""  Mutably borrows the value associated with the key in the bag bag: & mut ObjectBag. Aborts with suicdynamic field-EFieldToesNotExist if the bag does not have an entry with that key k: K. Aborts with suicdynamic field-EFieldToesNotExist if the bag has an entry for the key, but the value does not have the specified type. "bash ""  Mutably borrows the key-value pair in the bag bag: & mut ObjectBag and returns the value. Aborts with suicdynamic field-EFieldToesNotExist if the bag does not have an entry with that key k: K. Aborts with suicdynamic field-EFieldToesNotExist if the bag does not have an entry with that key k: K. Aborts with suicdynamic field-EFieldToesNotExist if the bag does not have an entry with that key k: K. Aborts with suicdynamic field-EFieldToesNotExist if the bag has an entry for the key, but the value does not have the specified type. "bash ""  Returns true iff there is an value associated with the key k: K in the bag bag: & ObjectBag with an assigned value of type V Returns true iff there is an value associated with the key k: K in the bag bag: & ObjectBag with an assigned value of type V	```bash
suicdynamic, field:EFieldTypeMismatch if the bag has an entry for the key, but the value does not have the specified type.  "bash  ""bash  ""B	
"bash "" Mutably borrows the value associated with the key in the bag bag: & mut ObjectBag. Aborts with suicdynamic field::FrieldTocsNotExist if the bag does not have an entry with that key k: K. Aborts with suicdynamic field::FrieldTypeMismatch if the bag has an entry for the key, but the value does not have the specified type. "bash "" Mutably borrows the key-value pair in the bag bag: & mut ObjectBag and returns the value. Aborts with suicdynamic field::FrieldTocsNotExist if the bag does not have an entry with that key k: K. Aborts with suicdynamic field::FrieldTypeMismatch if the bag has an entry for the key, but the value does not have the specified type. "bash "" Pash "" Returns true iff there is an value associated with the key k: K in the bag bag: & ObjectBag with an assigned value of type V  Returns true iff there is an value associated with the key k: K in the bag bag: & ObjectBag with an assigned value of type V	sui::dynamic_field::EFieldDoesNotExist if the bag does not have an entry with that key k: K. Aborts with
""bash "" Mutably borrows the value associated with the key in the bag bag: & mut ObjectBag. Aborts with suicdynamic_field::PFiekITopeMismatch if the bag does not have an entry with that key k: K. Aborts with suicdynamic_field::PFiekITopeMismatch if the bag has an entry for the key, but the value does not have the specified type. ""bash "" "bash "" Mutably borrows the key-value pair in the bag bag: & mut ObjectBag and returns the value. Aborts with suicdynamic_field::PFiekITopeMismatch if the bag does not have an entry with that key k: K. Aborts with suicdynamic_field::PFiekITopeMismatch if the bag has an entry for the key, but the value does not have the specified type. ""bash "" "bash "" Returns true iff there is an value associated with the key k: K in the bag bag: & ObjectBag with an assigned value of type V  Returns true iff there is an value associated with the key k: K in the bag bag: & ObjectBag with an assigned value of type V	```bash
Mutably borrows the value associated with the key in the bag bag: & mut ObjectBag. Aborts with suizdynamic_field::FFieldDoesNotExist if the bag does not have an entry with that key k: K. Aborts with suizdynamic_field::FFieldTypeMismatch if the bag has an entry for the key, but the value does not have the specified type.  "bash  "the bag bag: & mut ObjectBag and returns the value. Aborts with suizdynamic_field::FFieldTypeMismatch if the bag bag: & mut ObjectBag and returns the value. Aborts with suizdynamic_field::FFieldTypeMismatch if the bag has an entry for the key, but the value does not have the specified type.  "bash  "bash "charter iff there is an value associated with the key k: K in the bag bag: & ObjectBag "bash "charter iff there is an value associated with the key k: K in the bag bag: & ObjectBag with an assigned value of type V  Returns true iff there is an value associated with the key k: K in the bag bag: & ObjectBag with an assigned value of type V	
Mutably borrows the value associated with the key in the bag bag: & mut ObjectBag. Aborts with suizdynamic_field:FFieldDoesNotExist if the bag has an entry for the key, but the value does not have the specified type. "bash ""  Mutably borrows the key-value pair in the bag bag: & mut ObjectBag and returns the value. Aborts with suizdynamic_field:FFieldDoesNotExist if the bag bag: & mut ObjectBag and returns the value. Aborts with suizdynamic_field:FFieldDoesNotExist if the bag does not have an entry with that key k: K. Aborts with suizdynamic_field:FFieldDoesNotExist if the bag has an entry for the key, but the value does not have the specified type. "bash ""  Returns true iff there is an value associated with the key k: K in the bag bag: & ObjectBag with an assigned value of type V Returns true iff there is an value associated with the key k: K in the bag bag: & ObjectBag with an assigned value of type V	```bash
suizdynamic_field:EFieldTypeMismatch if the bag has an entry for the key, but the value does not have the specified type.  "bash  "the bag bag : & mut ObjectBag and returns the value. Aborts with suizdynamic_field:EFieldToesNotFxist if the bag bag : & mut ObjectBag and returns the value. Aborts with suizdynamic_field:EFieldToesNotFxist if the bag does not have an entry with that key k: K. Aborts with suizdynamic_field:EFieldToesNotFxist if the bag has an entry for the key, but the value does not have the specified type.  "bash  "charter if there is an value associated with the key k: K in the bag bag : & ObjectBag  "bash  "charter if there is an value associated with the key k: K in the bag bag : & ObjectBag  "bash  "charter if there is an value associated with the key k: K in the bag bag : & ObjectBag with an assigned value of type V	
""bash ""  Mutably borrows the key-value pair in the bag bag: & mut ObjectBag and returns the value. Aborts with suirdynamic_field:EFieldDoesNotExist if the bag does not have an entry with that key k: K. Aborts with suirdynamic_field:EFieldTypeMismatch if the bag has an entry for the key, but the value does not have the specified type. ""bash ""  Returns true iff there is an value associated with the key k: K in the bag bag: & ObjectBag ""bash ""  Returns true iff there is an value associated with the key k: K in the bag bag: & ObjectBag with an assigned value of type V	sui::dynamic_field::EFieldDoesNotExist if the bag does not have an entry with that key k: K. Aborts with
""bash ""  Mutably borrows the key-value pair in the bag bag: & mut ObjectBag and returns the value. Aborts with suixdynamic_field::EFieldToesNotExist if the bag does not have an entry with that key k: K. Aborts with suixdynamic_field::EFieldTypeMismatch if the bag has an entry for the key, but the value does not have the specified type. ""bash ""  Returns true iff there is an value associated with the key k: K in the bag bag: & ObjectBag ""bash ""  ""bash ""  Returns true iff there is an value associated with the key k: K in the bag bag: & ObjectBag with an assigned value of type V	```bash
Mutably borrows the key-value pair in the bag bag: & mut ObjectBag and returns the value. Aborts with suizdynamic field::EFieldDoesNotExist if the bag does not have an entry with that key k: K. Aborts with suizdynamic field::EFieldTypeMismatch if the bag has an entry for the key, but the value does not have the specified type.  ""bash ""  Returns true iff there is an value associated with the key k: K in the bag bag: & ObjectBag ""bash ""  Returns true iff there is an value associated with the key k: K in the bag bag: & ObjectBag with an assigned value of type V	
Mutably borrows the key-value pair in the bag bag: & mut ObjectBag and returns the value. Aborts with suixdynamic_field::EFieldDoesNotExist if the bag does not have an entry with that key k: K. Aborts with suixdynamic_field::EFieldTypeMismatch if the bag has an entry for the key, but the value does not have the specified type.  ""bash "" Returns true iff there is an value associated with the key k: K in the bag bag: & ObjectBag ""bash "" "bash "" Returns true iff there is an value associated with the key k: K in the bag bag: & ObjectBag with an assigned value of type V	```bash
sui::dynamic_field::EFieldDoesNotExist if the bag does not have an entry with that key k: K. Aborts with sui::dynamic_field::EFieldTypeMismatch if the bag has an entry for the key, but the value does not have the specified type.  ""bash "" Returns true iff there is an value associated with the key k: K in the bag bag: & ObjectBag ""bash "" ""bash "" Returns true iff there is an value associated with the key k: K in the bag bag: & ObjectBag with an assigned value of type V	
""bash ""bash shash ""bash ""bash ""bash "" "bash "" "bash "" "The shash	sui::dynamic_field::EFieldDoesNotExist if the bag does not have an entry with that key k: K. Aborts with
""bash ""bash ""bash ""bash "" ""bash "" ""bash "" ""bash "" ""bash "" Returns true iff there is an value associated with the key k: K in the bag bag: & ObjectBag with an assigned value of type V	```bash
Returns true iff there is an value associated with the key k: K in the bag bag: & ObjectBag  "bash "bash " Returns true iff there is an value associated with the key k: K in the bag bag: & ObjectBag with an assigned value of type V	
Returns true iff there is an value associated with the key k: K in the bag bag: & ObjectBag  "bash "bash " Returns true iff there is an value associated with the key k: K in the bag bag: & ObjectBag with an assigned value of type V	```bash
""bash ""bash "" Returns true iff there is an value associated with the key k: K in the bag bag: & ObjectBag with an assigned value of type V	
"bash "Returns true iff there is an value associated with the key k: K in the bag bag: & ObjectBag with an assigned value of type V	Returns true iff there is an value associated with the key k: K in the bag $\underline{bag}$ : & $\underline{ObjectBag}$
$\label{eq:continuous} \text{Returns true iff there is an value associated with the key $k$: $K$ in the bag $\underline{\text{bag}}$: & $\underline{\text{ObjectBag}}$ with an assigned value of type $V$}$	```bash
Returns true iff there is an value associated with the key k: $K$ in the bag $\underline{bag}$ : & $\underline{ObjectBag}$ with an assigned value of type $V$	
$\label{eq:continuous_equation} Returns \ true \ iff there \ is \ an \ value \ associated \ with \ the \ key \ k: K \ in \ the \ bag \ \underline{bag}: \& \ \underline{ObjectBag} \ with \ an \ assigned \ value \ of \ type \ V$	```bash
	Returns true iff there is an value associated with the key $k$ : $K$ in the bag $\underline{bag}$ : & $\underline{ObjectBag}$ with an assigned value of type $V$ ```bash

```
```bash
,,,
Returns the size of the bag, the number of key-value pairs
```bash
,,,
```bash
Returns true iff the bag is empty (if length returns 0)
```bash
```bash
Destroys an empty bag Aborts with EBagNotEmpty if the bag still contains values
```bash
```bash
Returns the ID of the object associated with the key if the bag has an entry with key k: K Returns none otherwise
```bash
```bash
Function
Immutably borrows the value associated with the key in the bag bag: & ObjectBag. Aborts with
sui::dynamic field::EFieldDoesNotExist if the bag does not have an entry with that key k: K. Aborts with
sui::dynamic field::EFieldTypeMismatch if the bag has an entry for the key, but the value does not have the specified type.
```bash
```bash
Mutably borrows the value associated with the key in the bag \underline{bag}: & mut \underline{ObjectBag}. Aborts with
```

sui::dynamic field::EFieldDoesNotExist if the bag does not have an entry with that key k: K. Aborts with

```bash

sui::dynamic field::EFieldTypeMismatch if the bag has an entry for the key, but the value does not have the specified type.

```
```bash
***
Mutably borrows the key-value pair in the bag bag: & mut ObjectBag and returns the value. Aborts with
sui::dynamic field::EFieldDoesNotExist if the bag does not have an entry with that key k: K. Aborts with
sui::dynamic field::EFieldTypeMismatch if the bag has an entry for the key, but the value does not have the specified type.
```bash
```bash
***
Returns true iff there is an value associated with the key k: K in the bag \underline{bag}: & \underline{ObjectBag}
```bash
...
```bash
Returns true iff there is an value associated with the key k: K in the bag bag: & ObjectBag with an assigned value of type V
```bash
```bash
Returns the size of the bag, the number of key-value pairs
```bash
***
```bash
Returns true iff the bag is empty (if length returns 0)
```bash
```bash
Destroys an empty bag Aborts with EBagNotEmpty if the bag still contains values
```bash
***
```bash
```

Returns the ID of the object associated with the key if the bag has an entry with key k: K Returns none otherwise
```bash
```bash
Function
Mutably borrows the value associated with the key in the bag $\underline{bag}$ : & mut $\underline{ObjectBag}$ . Aborts with $\underline{sui::dynamic\_field::EFieldDoesNotExist}$ if the bag does not have an entry with that key k: K. Aborts with $\underline{sui::dynamic\_field::EFieldTypeMismatch}$ if the bag has an entry for the key, but the value does not have the specified type.
```bash
```bash
Mutably borrows the key-value pair in the bag bag: & mut ObjectBag and returns the value. Aborts with sui::dynamic_field::EFieldDoesNotExist if the bag does not have an entry with that key k: K. Aborts with sui::dynamic_field::EFieldTypeMismatch if the bag has an entry for the key, but the value does not have the specified type.
```bash
···
```bash
Returns true iff there is an value associated with the key k: K in the bag $\underline{bag}$ : & $\underline{ObjectBag}$
```bash
```bash
Returns true iff there is an value associated with the key $k$ : $K$ in the bag $\underline{bag}$ : $\underline{\&}$ $\underline{ObjectBag}$ with an assigned value of type $V$
```bash
```bash
Returns the size of the bag, the number of key-value pairs
```bash
```bash

Returns true in the dag is empty (in <u>length</u> returns 0)
```bash
```bash
Destroys an empty bag Aborts with <a href="mailto:EBagNotEmpty">EBagNotEmpty</a> if the bag still contains values
```bash
```bash
Returns the ID of the object associated with the key if the bag has an entry with key k: K Returns none otherwise
```bash
```bash
Function
Mutably borrows the key-value pair in the bag bag: & mut ObjectBag and returns the value. Aborts with sui::dynamic_field::EFieldDoesNotExist if the bag does not have an entry with that key k: K. Aborts with sui::dynamic_field::EFieldTypeMismatch if the bag has an entry for the key, but the value does not have the specified type.
```bash
```bash
Returns true iff there is an value associated with the key k: K in the bag bag: & ObjectBag
```bash
```bash
Returns true iff there is an value associated with the key k: K in the bag bag: & ObjectBag with an assigned value of type V
```bash
```bash

Returns the size of the bag, the number of key-value pairs

```
```bash
***
```bash
Returns true iff the bag is empty (if length returns 0)
```bash
```bash
Destroys an empty bag Aborts with <a href="EBagNotEmpty">EBagNotEmpty</a> if the bag still contains values
```bash
***
```bash
Returns the ID of the object associated with the key if the bag has an entry with key k: K Returns none otherwise
```bash
```bash
Function
Returns true iff there is an value associated with the key k: K in the bag \underline{bag}: & \underline{ObjectBag}
```bash
```bash
Returns true iff there is an value associated with the key k: K in the bag bag: & ObjectBag with an assigned value of type V
```bash
***
```bash
Returns the size of the bag, the number of key-value pairs
```bash
```bash
```

```
Returns true iff the bag is empty (if length returns 0)
```bash
```bash
***
Destroys an empty bag Aborts with EBagNotEmpty if the bag still contains values
```bash
```bash
***
Returns the ID of the object associated with the key if the bag has an entry with key k: K Returns none otherwise
```bash
```bash
Function
Returns true iff there is an value associated with the key k: K in the bag <u>bag</u>: & <u>ObjectBag</u> with an assigned value of type V
```bash
```bash
Returns the size of the bag, the number of key-value pairs
```bash
***
```bash
Returns true iff the bag is empty (if length returns 0)
```bash
```bash
Destroys an empty bag Aborts with <a href="mailto:EBagNotEmpty">EBagNotEmpty</a> if the bag still contains values
```bash
```

```
```bash
...
Returns the ID of the object associated with the key if the bag has an entry with key k: K Returns none otherwise
```bash
***
```bash
Function
Returns the size of the bag, the number of key-value pairs
```bash
***
```bash
***
Returns true iff the bag is empty (if length returns 0)
```bash
```bash
Destroys an empty bag Aborts with <a href="EBagNotEmpty">EBagNotEmpty</a> if the bag still contains values
```bash
***
```bash
Returns the ID of the object associated with the key if the bag has an entry with key k: K Returns none otherwise
```bash
```bash
Function
Returns true iff the bag is empty (if length returns 0)
```bash
,,,
```

```bash
Destroys an empty bag Aborts with <u>EBagNotEmpty</u> if the bag still contains values
```bash
```bash
···
Returns the ID of the object associated with the key if the bag has an entry with key k: K Returns none otherwise
```bash
```bash
···
Function
Destroys an empty bag Aborts with <u>EBagNotEmpty</u> if the bag still contains values
```bash
···
```bash
···
Returns the ID of the object associated with the key if the bag has an entry with key k: K Returns none otherwise
```bash
···
```bash
Function
Returns the ID of the object associated with the key if the bag has an entry with key k: K Returns none otherwise
```bash
```bash