附录 V. 泛型具象

表V-1. 内置类型的泛型具象

类型	等价泛型具象
tuple[*T*]	typing.Tuple[*T*, ...]
list[*T*]	*class* typing.List(list, MutableSequence[*T*])
dict[*KT*, *VT*]	*class* typing.Dict(dict, MutableMapping[*KT*, *VT*])
set[*T*]	*class* typing.Set(list, MutableSet[*T*])
frozenset[*T_co*]	*class* typing.FrozenSet(frozenset, AbstractSet[*T_co*])

表V-2. collections模块的泛型具象

类型	等价泛型具象
deque[*T*]	*class* typing.Deque(deque, MutableSequence[*T*])
ChainMap[*KT*, *VT*]	*class* typing.ChainMap(ChainMap, MutableMapping[*KT*, *VT*])
Counter[*T*, *int*]	*class* typing.Counter(Counter, Dict[*T*, *int*])
OrderedDict[*KT*, *VT*]	*class* typing.OrderedDict(OrderedDict, MutableMapping[*KT*, *VT*])
defaultdict[*KT*, *VT*]	*class* typing.DefaultDict(defaultdict, MutableMapping[*KT*, *VT*])

表V-3. collections.abc模块的泛型具象

类型	等价泛型具象
Sized	*class* typing.Sized
Container[*T_co*]	*class* typing.Container(Generic[*T_co*])
Iterable[*T_co*]	*class* typing.Iterable(Generic[*T_co*])
Iterator[*T_co*]	*class* typing.Container(Iterable[*T_co*])
Generator[*T_co*, *T_contra*, *V_co*]	*class* typing.Generator(Iterator[*T_co*], Generic[*T_co*, *T_contra*, *V_co*])
Reversible[*T_co*]	*class* typing.Reversible(Iterable[*T_co*])
Collection[*T_co*]	*class* typing.Collection(Sized, Iterable[*T_co*], Container[*T_co*])
Sequence[*T_co*]	*class* typing.Sequence(Reversible[*T_co*], Collection[*T_co*])
MutableSequence[*T*]	*class* typing.MutableSequence(Sequence[*T*])
ByteString[*int*]	*class* typing.ByteString(Sequence[*int*])

类型	等价泛型具象
Mapping[*KT*, *VT_co*]	*class* typing.Mapping(Sized, Collection[*KT*], Generic[*VT_co*])
MutableMapping[*KT*, *VT*]	*class* typing.MutableMapping(Mapping[*KT*, *VT*])
Set[*T_co*]	*class* typing.AbstractSet(Sized, Collection[*T_co*])
MutableSet[*T*]	*class* typing.MutableSet(AbstractSet[*T*])
MappingView[*T_co*]	*class* typing.MappingView(Sized, Iterable[*T_co*])
ItemsView[*KT_co*, *VT_co*]	*class* typing.ItemsView(MappingView, Generic[*KT_co*, *VT_co*])
KeysView[*KT_co*]	*class* typing.KeysView(MappingView[*KT_co*], AbstractSet[*KT_co*])
ValuesView[*VT_co*]	*class* typing.ValuesView(MappingView[*VT_co*])
Hashable	*class* typing.Hashable
Callable[ATs_contra*, *RT_co*]**	Callable[[**ATs_contra*], *RT_co*]
Awaitable[*T_co*]	*class* typing.Awaitable(Generic[*T_co*])
Coroutine[*T_co*, *T_contra*, *V_co*]	*class* typing.Coroutine(Awaitable[*V_co*], Generic[*T_co*, *T_contra*, *V_co*])
AsyncIterable[*T_co*]	*class* typing.AsyncIterable(Generic[*T_co*])
AsyncIterator[*T_co*]	*class* typing.AsyncIterator(AsyncIterable[*T_co*])
AsyncGenerator[*T_co*, *T_contra*]	*class* typing.AsyncGenerator(AsyncIterator[*T_co*], Generic[*T_co*, *T_contra*])

表V-4. contextlib模块的泛型具象

类型	等价泛型具象
AbstractContextManager[*T_co*]	*class* typing.ContextManager(Generic[*T_co*])
AbstractAsyncContextManager[*T_co*]	*class* typing.AsyncContextManager(Generic[*T_co*])

表V-5. io模块的泛型具象

类型	等价泛型具象
io[typing.AnyStr]	*class* typing.IO
binaryio[IO[bytes]]	*class* typing.BinaryIO
textio[IO[str]]	*class* typing.TextIO

表V-6. re模块的泛型具象

类型	等价泛型具象
Pattern[typing.AnyStr]	*class* typing.Pattern
Match[typing.AnyStr]	*class* typing.Match