

Namespace BOOSE_WebApplication. Controllers

Classes

[CanvasController](#)

[HomeController](#)

Class CanvasController

Namespace: [BOOSE_WebApplication.Controllers](#)

Assembly: BOOSE_WebApplication.dll

```
public class CanvasController : Controller, IActionFilter, IAsyncActionFilter,  
IFilterMetadata, IDisposable
```

Inheritance

[object](#) ← [ControllerBase](#) ← [Controller](#) ← [CanvasController](#)

Implements

[IActionFilter](#), [IAsyncActionFilter](#), [IFilterMetadata](#), [IDisposable](#)

Inherited Members

[Controller.View\(\)](#), [Controller.View\(string\)](#), [Controller.View\(object\)](#),
[Controller.View\(string, object\)](#), [Controller.PartialView\(\)](#), [Controller.PartialView\(string\)](#),
[Controller.PartialView\(object\)](#), [Controller.PartialView\(string, object\)](#),
[Controller.ViewComponent\(string\)](#), [Controller.ViewComponent\(Type\)](#),
[Controller.ViewComponent\(string, object\)](#), [Controller.ViewComponent\(Type, object\)](#),
[Controller.Json\(object\)](#), [Controller.Json\(object, object\)](#),
[Controller.OnActionExecuting\(ActionExecutingContext\)](#),
[Controller.OnActionExecuted\(ActionExecutedContext\)](#),
[Controller.OnActionExecutionAsync\(ActionExecutingContext, ActionExecutionDelegate\)](#),
[Controller.Dispose\(\)](#), [Controller.Dispose\(bool\)](#), [Controller.ViewData](#), [Controller.TempData](#),
[Controller.ViewBag](#), [ControllerBase.StatusCode\(int\)](#), [ControllerBase.StatusCode\(int, object\)](#),
[ControllerBase.Content\(string\)](#), [ControllerBase.Content\(string, string\)](#),
[ControllerBase.Content\(string, string, Encoding\)](#),
[ControllerBase.Content\(string, MediaTypeHeaderValue\)](#), [ControllerBase.NoContent\(\)](#),
[ControllerBase.Ok\(\)](#), [ControllerBase.Ok\(object\)](#), [ControllerBase.Redirect\(string\)](#),
[ControllerBase.RedirectPermanent\(string\)](#), [ControllerBase.RedirectPreserveMethod\(string\)](#),
[ControllerBase.RedirectPermanentPreserveMethod\(string\)](#), [ControllerBase.LocalRedirect\(string\)](#),
[ControllerBase.LocalRedirectPermanent\(string\)](#), [ControllerBase.LocalRedirectPreserveMethod\(string\)](#),
[ControllerBase.LocalRedirectPermanentPreserveMethod\(string\)](#), [ControllerBase.RedirectToAction\(\)](#),
[ControllerBase.RedirectToAction\(string\)](#), [ControllerBase.RedirectToAction\(string, object\)](#),
[ControllerBase.RedirectToAction\(string, string\)](#),
[ControllerBase.RedirectToAction\(string, string, object\)](#),
[ControllerBase.RedirectToAction\(string, string, string\)](#),
[ControllerBase.RedirectToAction\(string, string, object, string\)](#),

[ControllerBase.RedirectToActionPreserveMethod\(string, string, object, string\)](#) ,
 [ControllerBase.RedirectToActionPermanent\(string\)](#) ,
 [ControllerBase.RedirectToActionPermanent\(string, object\)](#) ,
 [ControllerBase.RedirectToActionPermanent\(string, string\)](#) ,
 [ControllerBase.RedirectToActionPermanent\(string, string, string\)](#) ,
 [ControllerBase.RedirectToActionPermanent\(string, string, object\)](#) ,
 [ControllerBase.RedirectToActionPermanent\(string, string, object, string\)](#) ,
 [ControllerBase.RedirectToActionPermanentPreserveMethod\(string, string, object, string\)](#) ,
 [ControllerBase.RedirectToRoute\(string\)](#) , ControllerBase.RedirectToRoute(object) ,
 [ControllerBase.RedirectToRoute\(string, object\)](#) , ControllerBase.RedirectToRoute(string, string) ,
 [ControllerBase.RedirectToRoute\(string, object, string\)](#) ,
 [ControllerBase.RedirectToRoutePreserveMethod\(string, object, string\)](#) ,
 [ControllerBase.RedirectToRoutePermanent\(string\)](#) ,
 [ControllerBase.RedirectToRoutePermanent\(object\)](#) ,
 [ControllerBase.RedirectToRoutePermanent\(string, object\)](#) ,
 [ControllerBase.RedirectToRoutePermanent\(string, string\)](#) ,
 [ControllerBase.RedirectToRoutePermanent\(string, object, string\)](#) ,
 [ControllerBase.RedirectToRoutePermanentPreserveMethod\(string, object, string\)](#) ,
 [ControllerBase.RedirectToPage\(string\)](#) , ControllerBase.RedirectToPage(string, object) ,
 [ControllerBase.RedirectToPage\(string, string\)](#) , ControllerBase.RedirectToPage(string, string, object) ,
 [ControllerBase.RedirectToPage\(string, string, string\)](#) ,
 [ControllerBase.RedirectToPage\(string, string, object, string\)](#) ,
 [ControllerBase.RedirectToPagePermanent\(string\)](#) ,
 [ControllerBase.RedirectToPagePermanent\(string, object\)](#) ,
 [ControllerBase.RedirectToPagePermanent\(string, string\)](#) ,
 [ControllerBase.RedirectToPagePermanent\(string, string, string\)](#) ,
 [ControllerBase.RedirectToPagePermanent\(string, string, object, string\)](#) ,
 [ControllerBase.RedirectToPagePreserveMethod\(string, string, object, string\)](#) ,
 [ControllerBase.RedirectToPagePermanentPreserveMethod\(string, string, object, string\)](#) ,
 [ControllerBase.File\(byte\[\], string\)](#) , ControllerBase.File(byte[], string, bool) ,
 [ControllerBase.File\(byte\[\], string, string\)](#) , ControllerBase.File(byte[], string, string, bool) ,
 [ControllerBase.File\(byte\[\], string, DateTimeOffset?, EntityTagHeaderValue\)](#) ,
 [ControllerBase.File\(byte\[\], string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#) ,
 [ControllerBase.File\(byte\[\], string, string, DateTimeOffset?, EntityTagHeaderValue\)](#) ,
 [ControllerBase.File\(byte\[\], string, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#) ,
 [ControllerBase.File\(Stream, string\)](#) , ControllerBase.File(Stream, string, bool) ,
 [ControllerBase.File\(Stream, string, string\)](#) , ControllerBase.File(Stream, string, string, bool) ,
 [ControllerBase.File\(Stream, string, DateTimeOffset?\)](#) , EntityTagHeaderValue ,
 [ControllerBase.File\(Stream, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#) ,
 [ControllerBase.File\(Stream, string, string, DateTimeOffset?, EntityTagHeaderValue\)](#) ,

[ControllerBase.File\(Stream, string, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#) ,
 [ControllerBase.File\(string, string\)](#) , [ControllerBase.File\(string, string, bool\)](#) ,
 [ControllerBase.File\(string, string, string\)](#) , [ControllerBase.File\(string, string, string, bool\)](#) ,
 [ControllerBase.File\(string, string, DateTimeOffset?, EntityTagHeaderValue\)](#) ,
 [ControllerBase.File\(string, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#) ,
 [ControllerBase.File\(string, string, string, DateTimeOffset?, EntityTagHeaderValue\)](#) ,
 [ControllerBase.File\(string, string, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#) ,
 [ControllerBase.PhysicalFile\(string, string\)](#) , [ControllerBase.PhysicalFile\(string, string, bool\)](#) ,
 [ControllerBase.PhysicalFile\(string, string, string\)](#) ,
 [ControllerBase.PhysicalFile\(string, string, string, bool\)](#) ,
 [ControllerBase.PhysicalFile\(string, string, DateTimeOffset?, EntityTagHeaderValue\)](#) ,
 [ControllerBase.PhysicalFile\(string, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#) ,
 [ControllerBase.PhysicalFile\(string, string, string, DateTimeOffset?, EntityTagHeaderValue\)](#) ,
 [ControllerBase.PhysicalFile\(string, string, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#) ,
 [ControllerBase.Unauthorized\(\)](#) , [ControllerBase.Unauthorized\(object\)](#) , [ControllerBase.NotFound\(\)](#) ,
 [ControllerBase.NotFound\(object\)](#) , [ControllerBase.BadRequest\(\)](#) ,
 [ControllerBase.BadRequest\(object\)](#) , [ControllerBase.BadRequest\(ModelStateDictionary\)](#) ,
 [ControllerBase.UnprocessableEntity\(\)](#) , [ControllerBase.UnprocessableEntity\(object\)](#) ,
 [ControllerBase.UnprocessableEntity\(ModelStateDictionary\)](#) , [ControllerBase.Conflict\(\)](#) ,
 [ControllerBase.Conflict\(object\)](#) , [ControllerBase.Conflict\(ModelStateDictionary\)](#) ,
 [ControllerBase.Problem\(string, string, int?, string, string\)](#) ,
 [ControllerBase.ValidationProblem\(ValidationProblemDetails\)](#) ,
 [ControllerBase.ValidationProblem\(ModelStateDictionary\)](#) , [ControllerBase.ValidationProblem\(\)](#) ,
 [ControllerBase.ValidationProblem\(string, string, int?, string, string, ModelStateDictionary\)](#) ,
 [ControllerBase.Created\(\)](#) , [ControllerBase.Created\(string, object\)](#) ,
 [ControllerBase.Created\(Uri, object\)](#) , [ControllerBase.CreatedAtAction\(string, object\)](#) ,
 [ControllerBase.CreatedAtAction\(string, object, object\)](#) ,
 [ControllerBase.CreatedAtAction\(string, string, object, object\)](#) ,
 [ControllerBase.CreatedAtRoute\(string, object\)](#) , [ControllerBase.CreatedAtRoute\(object, object\)](#) ,
 [ControllerBase.CreatedAtRoute\(string, object, object\)](#) , [ControllerBase.Accepted\(\)](#) ,
 [ControllerBase.Accepted\(object\)](#) , [ControllerBase.Accepted\(Uri\)](#) , [ControllerBase.Accepted\(string\)](#) ,
 [ControllerBase.Accepted\(string, object\)](#) , [ControllerBase.Accepted\(Uri, object\)](#) ,
 [ControllerBase.AcceptedAtAction\(string\)](#) , [ControllerBase.AcceptedAtAction\(string, string\)](#) ,
 [ControllerBase.AcceptedAtAction\(string, object\)](#) ,
 [ControllerBase.AcceptedAtAction\(string, string, object\)](#) ,
 [ControllerBase.AcceptedAtAction\(string, object, object\)](#) ,
 [ControllerBase.AcceptedAtAction\(string, string, object, object\)](#) ,
 [ControllerBase.AcceptedAtRoute\(object\)](#) , [ControllerBase.AcceptedAtRoute\(string\)](#) ,
 [ControllerBase.AcceptedAtRoute\(string, object\)](#) , [ControllerBase.AcceptedAtRoute\(object, object\)](#) ,
 [ControllerBase.AcceptedAtRoute\(string, object, object\)](#) , [ControllerBase.Challenge\(\)](#) ,

[ControllerBase.Challenge\(params string\[\]\)](#) , [ControllerBase.Challenge\(AuthenticationProperties\)](#) ,
 [ControllerBase.Challenge\(AuthenticationProperties, params string\[\]\)](#) , [ControllerBase.Forbid\(\)](#) ,
 [ControllerBase.Forbid\(params string\[\]\)](#) , [ControllerBase.Forbid\(AuthenticationProperties\)](#) ,
 [ControllerBase.Forbid\(AuthenticationProperties, params string\[\]\)](#) ,
 [ControllerBase.SignIn\(ClaimsPrincipal\)](#) , [ControllerBase.SignIn\(ClaimsPrincipal, string\)](#) ,
 [ControllerBase.SignIn\(ClaimsPrincipal, AuthenticationProperties\)](#) ,
 [ControllerBase.SignIn\(ClaimsPrincipal, AuthenticationProperties, string\)](#) , [ControllerBase.SignOut\(\)](#) ,
 [ControllerBase.SignOut\(AuthenticationProperties\)](#) , [ControllerBase.SignOut\(params string\[\]\)](#) ,
 [ControllerBase.SignOut\(AuthenticationProperties, params string\[\]\)](#) ,
 [ControllerBase.TryUpdateModelAsync<TModel>\(TModel\)](#) ,
 [ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string\)](#) ,
 [ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string, IValueProvider\)](#) ,
 [ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string, params Expression<Func<TModel, object>>\[\]\)](#) ,
 [ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string, Func<ModelMetadata, bool>\)](#) ,
 [ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string, IValueProvider, params Expression<Func<TModel, object>>\[\]\)](#) ,
 [ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string, IValueProvider, Func<ModelMetadata, bool>\)](#) ,
 [ControllerBase.TryUpdateModelAsync\(object, Type, string\)](#) ,
 [ControllerBase.TryUpdateModelAsync\(object, Type, string, IValueProvider, Func<ModelMetadata, bool>\)](#) ,
 [ControllerBase.TryValidateModel\(object\)](#) , [ControllerBase.TryValidateModel\(object, string\)](#) ,
 [ControllerBase.HttpContext](#) , [ControllerBase.Request](#) , [ControllerBase.Response](#) ,
 [ControllerBase.RouteData](#) , [ControllerBase.ModelState](#) , [ControllerBase.ControllerContext](#) ,
 [ControllerBase.MetadataProvider](#) , [ControllerBase.ModelBinderFactory](#) , [ControllerBase.Url](#) ,
 [ControllerBase.ObjectValidator](#) , [ControllerBase.ProblemDetailsFactory](#) , [ControllerBase.User](#) ,
 [ControllerBase.Empty](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
 [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) ,
 [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Constructors

CanvasController()

Initializes the CanvasController with necessary dependencies.

```
public CanvasController()
```

Methods

CanvasView()

Displays the view for the canvas.

```
public ActionResult CanvasView()
```

Returns

[ActionResult](#)

ClearCanvas()

Clears the canvas and resets the program state.

```
[HttpPost]  
public JsonResult ClearCanvas()
```

Returns

[JsonResult](#)

JSON response indicating success or failure.

RunProgram(JsonElement)

Executes the program provided by the user and updates the canvas.

```
[HttpPost]  
public JsonResult RunProgram(JsonElement payload)
```

Parameters

`payload` [JsonElement](#)

Returns

[JsonResult](#)

JSON response with success status and updated canvas as a base64 image.

Class HomeController

Namespace: [BOOSE_WebApplication.Controllers](#)

Assembly: BOOSE_WebApplication.dll

```
public class HomeController : Controller, IActionFilter, IAsyncActionFilter,  
IFilterMetadata, IDisposable
```

Inheritance

[object](#) ← [ControllerBase](#) ← [Controller](#) ← HomeController

Implements

[IActionFilter](#), [IAsyncActionFilter](#), [IFilterMetadata](#), [IDisposable](#)

Inherited Members

[Controller.View\(\)](#), [Controller.View\(string\)](#), [Controller.View\(object\)](#),
[Controller.View\(string, object\)](#), [Controller.PartialView\(\)](#), [Controller.PartialView\(string\)](#),
[Controller.PartialView\(object\)](#), [Controller.PartialView\(string, object\)](#),
[Controller.ViewComponent\(string\)](#), [Controller.ViewComponent\(Type\)](#),
[Controller.ViewComponent\(string, object\)](#), [Controller.ViewComponent\(Type, object\)](#),
[Controller.Json\(object\)](#), [Controller.Json\(object, object\)](#),
[Controller.OnActionExecuting\(ActionExecutingContext\)](#),
[Controller.OnActionExecuted\(ActionExecutedContext\)](#),
[Controller.OnActionExecutionAsync\(ActionExecutingContext, ActionExecutionDelegate\)](#),
[Controller.Dispose\(\)](#), [Controller.Dispose\(bool\)](#), [Controller.ViewData](#), [Controller.TempData](#),
[Controller.ViewBag](#), [ControllerBase.StatusCode\(int\)](#), [ControllerBase.StatusCode\(int, object\)](#),
[ControllerBase.Content\(string\)](#), [ControllerBase.Content\(string, string\)](#),
[ControllerBase.Content\(string, string, Encoding\)](#),
[ControllerBase.Content\(string, MediaTypeHeaderValue\)](#), [ControllerBase.NoContent\(\)](#),
[ControllerBase.Ok\(\)](#), [ControllerBase.Ok\(object\)](#), [ControllerBase.Redirect\(string\)](#),
[ControllerBase.RedirectPermanent\(string\)](#), [ControllerBase.RedirectPreserveMethod\(string\)](#),
[ControllerBase.RedirectPermanentPreserveMethod\(string\)](#), [ControllerBase.LocalRedirect\(string\)](#),
[ControllerBase.LocalRedirectPermanent\(string\)](#), [ControllerBase.LocalRedirectPreserveMethod\(string\)](#),
[ControllerBase.LocalRedirectPermanentPreserveMethod\(string\)](#), [ControllerBase.RedirectToAction\(\)](#),
[ControllerBase.RedirectToAction\(string\)](#), [ControllerBase.RedirectToAction\(string, object\)](#),
[ControllerBase.RedirectToAction\(string, string\)](#),
[ControllerBase.RedirectToAction\(string, string, object\)](#),
[ControllerBase.RedirectToAction\(string, string, string\)](#),
[ControllerBase.RedirectToAction\(string, string, object, string\)](#),

[ControllerBase.RedirectToActionPreserveMethod\(string, string, object, string\)](#) ,
 [ControllerBase.RedirectToActionPermanent\(string\)](#) ,
 [ControllerBase.RedirectToActionPermanent\(string, object\)](#) ,
 [ControllerBase.RedirectToActionPermanent\(string, string\)](#) ,
 [ControllerBase.RedirectToActionPermanent\(string, string, string\)](#) ,
 [ControllerBase.RedirectToActionPermanent\(string, string, object\)](#) ,
 [ControllerBase.RedirectToActionPermanent\(string, string, object, string\)](#) ,
 [ControllerBase.RedirectToActionPermanentPreserveMethod\(string, string, object, string\)](#) ,
 [ControllerBase.RedirectToRoute\(string\)](#) , ControllerBase.RedirectToRoute(object) ,
 [ControllerBase.RedirectToRoute\(string, object\)](#) , ControllerBase.RedirectToRoute(string, string) ,
 [ControllerBase.RedirectToRoute\(string, object, string\)](#) ,
 [ControllerBase.RedirectToRoutePreserveMethod\(string, object, string\)](#) ,
 [ControllerBase.RedirectToRoutePermanent\(string\)](#) ,
 [ControllerBase.RedirectToRoutePermanent\(object\)](#) ,
 [ControllerBase.RedirectToRoutePermanent\(string, object\)](#) ,
 [ControllerBase.RedirectToRoutePermanent\(string, string\)](#) ,
 [ControllerBase.RedirectToRoutePermanent\(string, object, string\)](#) ,
 [ControllerBase.RedirectToRoutePermanentPreserveMethod\(string, object, string\)](#) ,
 [ControllerBase.RedirectToPage\(string\)](#) , ControllerBase.RedirectToPage(string, object) ,
 [ControllerBase.RedirectToPage\(string, string\)](#) , ControllerBase.RedirectToPage(string, string, object) ,
 [ControllerBase.RedirectToPage\(string, string, string\)](#) ,
 [ControllerBase.RedirectToPage\(string, string, object, string\)](#) ,
 [ControllerBase.RedirectToPagePermanent\(string\)](#) ,
 [ControllerBase.RedirectToPagePermanent\(string, object\)](#) ,
 [ControllerBase.RedirectToPagePermanent\(string, string\)](#) ,
 [ControllerBase.RedirectToPagePermanent\(string, string, string\)](#) ,
 [ControllerBase.RedirectToPagePermanent\(string, string, object, string\)](#) ,
 [ControllerBase.RedirectToPagePreserveMethod\(string, string, object, string\)](#) ,
 [ControllerBase.RedirectToPagePermanentPreserveMethod\(string, string, object, string\)](#) ,
 [ControllerBase.File\(byte\[\], string\)](#) , ControllerBase.File(byte[], string, bool) ,
 [ControllerBase.File\(byte\[\], string, string\)](#) , ControllerBase.File(byte[], string, string, bool) ,
 [ControllerBase.File\(byte\[\], string, DateTimeOffset?, EntityTagHeaderValue\)](#) ,
 [ControllerBase.File\(byte\[\], string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#) ,
 [ControllerBase.File\(byte\[\], string, string, DateTimeOffset?, EntityTagHeaderValue\)](#) ,
 [ControllerBase.File\(byte\[\], string, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#) ,
 [ControllerBase.File\(Stream, string\)](#) , ControllerBase.File(Stream, string, bool) ,
 [ControllerBase.File\(Stream, string, string\)](#) , ControllerBase.File(Stream, string, string, bool) ,
 [ControllerBase.File\(Stream, string, DateTimeOffset?\)](#) , EntityTagHeaderValue ,
 [ControllerBase.File\(Stream, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#) ,
 [ControllerBase.File\(Stream, string, string, DateTimeOffset?, EntityTagHeaderValue\)](#) ,

[ControllerBase.File\(Stream, string, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#) ,
 [ControllerBase.File\(string, string\)](#) , [ControllerBase.File\(string, string, bool\)](#) ,
 [ControllerBase.File\(string, string, string\)](#) , [ControllerBase.File\(string, string, string, bool\)](#) ,
 [ControllerBase.File\(string, string, DateTimeOffset?, EntityTagHeaderValue\)](#) ,
 [ControllerBase.File\(string, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#) ,
 [ControllerBase.File\(string, string, string, DateTimeOffset?, EntityTagHeaderValue\)](#) ,
 [ControllerBase.File\(string, string, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#) ,
 [ControllerBase.PhysicalFile\(string, string\)](#) , [ControllerBase.PhysicalFile\(string, string, bool\)](#) ,
 [ControllerBase.PhysicalFile\(string, string, string\)](#) ,
 [ControllerBase.PhysicalFile\(string, string, string, bool\)](#) ,
 [ControllerBase.PhysicalFile\(string, string, DateTimeOffset?, EntityTagHeaderValue\)](#) ,
 [ControllerBase.PhysicalFile\(string, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#) ,
 [ControllerBase.PhysicalFile\(string, string, string, DateTimeOffset?, EntityTagHeaderValue\)](#) ,
 [ControllerBase.PhysicalFile\(string, string, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#) ,
 [ControllerBase.Unauthorized\(\)](#) , [ControllerBase.Unauthorized\(object\)](#) , [ControllerBase.NotFound\(\)](#) ,
 [ControllerBase.NotFound\(object\)](#) , [ControllerBase.BadRequest\(\)](#) ,
 [ControllerBase.BadRequest\(object\)](#) , [ControllerBase.BadRequest\(ModelStateDictionary\)](#) ,
 [ControllerBase.UnprocessableEntity\(\)](#) , [ControllerBase.UnprocessableEntity\(object\)](#) ,
 [ControllerBase.UnprocessableEntity\(ModelStateDictionary\)](#) , [ControllerBase.Conflict\(\)](#) ,
 [ControllerBase.Conflict\(object\)](#) , [ControllerBase.Conflict\(ModelStateDictionary\)](#) ,
 [ControllerBase.Problem\(string, string, int?, string, string\)](#) ,
 [ControllerBase.ValidationProblem\(ValidationProblemDetails\)](#) ,
 [ControllerBase.ValidationProblem\(ModelStateDictionary\)](#) , [ControllerBase.ValidationProblem\(\)](#) ,
 [ControllerBase.ValidationProblem\(string, string, int?, string, string, ModelStateDictionary\)](#) ,
 [ControllerBase.Created\(\)](#) , [ControllerBase.Created\(string, object\)](#) ,
 [ControllerBase.Created\(Uri, object\)](#) , [ControllerBase.CreatedAtAction\(string, object\)](#) ,
 [ControllerBase.CreatedAtAction\(string, object, object\)](#) ,
 [ControllerBase.CreatedAtAction\(string, string, object, object\)](#) ,
 [ControllerBase.CreatedAtRoute\(string, object\)](#) , [ControllerBase.CreatedAtRoute\(object, object\)](#) ,
 [ControllerBase.CreatedAtRoute\(string, object, object\)](#) , [ControllerBase.Accepted\(\)](#) ,
 [ControllerBase.Accepted\(object\)](#) , [ControllerBase.Accepted\(Uri\)](#) , [ControllerBase.Accepted\(string\)](#) ,
 [ControllerBase.Accepted\(string, object\)](#) , [ControllerBase.Accepted\(Uri, object\)](#) ,
 [ControllerBase.AcceptedAtAction\(string\)](#) , [ControllerBase.AcceptedAtAction\(string, string\)](#) ,
 [ControllerBase.AcceptedAtAction\(string, object\)](#) ,
 [ControllerBase.AcceptedAtAction\(string, string, object\)](#) ,
 [ControllerBase.AcceptedAtAction\(string, object, object\)](#) ,
 [ControllerBase.AcceptedAtAction\(string, string, object, object\)](#) ,
 [ControllerBase.AcceptedAtRoute\(object\)](#) , [ControllerBase.AcceptedAtRoute\(string\)](#) ,
 [ControllerBase.AcceptedAtRoute\(string, object\)](#) , [ControllerBase.AcceptedAtRoute\(object, object\)](#) ,
 [ControllerBase.AcceptedAtRoute\(string, object, object\)](#) , [ControllerBase.Challenge\(\)](#) ,

[ControllerBase.Challenge\(params string\[\]\)](#) , [ControllerBase.Challenge\(AuthenticationProperties\)](#) ,
 [ControllerBase.Challenge\(AuthenticationProperties, params string\[\]\)](#) , [ControllerBase.Forbid\(\)](#) ,
 [ControllerBase.Forbid\(params string\[\]\)](#) , [ControllerBase.Forbid\(AuthenticationProperties\)](#) ,
 [ControllerBase.Forbid\(AuthenticationProperties, params string\[\]\)](#) ,
 [ControllerBase.SignIn\(ClaimsPrincipal\)](#) , [ControllerBase.SignIn\(ClaimsPrincipal, string\)](#) ,
 [ControllerBase.SignIn\(ClaimsPrincipal, AuthenticationProperties\)](#) ,
 [ControllerBase.SignIn\(ClaimsPrincipal, AuthenticationProperties, string\)](#) , [ControllerBase.SignOut\(\)](#) ,
 [ControllerBase.SignOut\(AuthenticationProperties\)](#) , [ControllerBase.SignOut\(params string\[\]\)](#) ,
 [ControllerBase.SignOut\(AuthenticationProperties, params string\[\]\)](#) ,
 [ControllerBase.TryUpdateModelAsync<TModel>\(TModel\)](#) ,
 [ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string\)](#) ,
 [ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string, IValueProvider\)](#) ,
 [ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string, params Expression<Func<TModel, object>>\[\]\)](#) ,
 [ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string, Func<ModelMetadata, bool>\)](#) ,
 [ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string, IValueProvider, params Expression<Func<TModel, object>>\[\]\)](#) ,
 [ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string, IValueProvider, Func<ModelMetadata, bool>\)](#) ,
 [ControllerBase.TryUpdateModelAsync\(object, Type, string\)](#) ,
 [ControllerBase.TryUpdateModelAsync\(object, Type, string, IValueProvider, Func<ModelMetadata, bool>\)](#) ,
 [ControllerBase.TryValidateModel\(object\)](#) , [ControllerBase.TryValidateModel\(object, string\)](#) ,
 [ControllerBase.HttpContext](#) , [ControllerBase.Request](#) , [ControllerBase.Response](#) ,
 [ControllerBase.RouteData](#) , [ControllerBase.ModelState](#) , [ControllerBase.ControllerContext](#) ,
 [ControllerBase.MetadataProvider](#) , [ControllerBase.ModelBinderFactory](#) , [ControllerBase.Url](#) ,
 [ControllerBase.ObjectValidator](#) , [ControllerBase.ProblemDetailsFactory](#) , [ControllerBase.User](#) ,
 [ControllerBase.Empty](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
 [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) ,
 [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Constructors

HomeController(ILocator<HomeController>)

```
public HomeController(ILocator<HomeController> logger)
```

Parameters

```
logger ILogger<HomeController>
```

Methods

Error()

```
[ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore = true)]  
public IActionResult Error()
```

Returns

[IActionResult](#)

Index()

```
public IActionResult Index()
```

Returns

[IActionResult](#)

Privacy()

```
public IActionResult Privacy()
```

Returns

[IActionResult](#)

Namespace BOOSE_WebApplication.Models

Classes

[ErrorViewModel](#)

Class ErrorViewModel

Namespace: [BOOSE_WebApplication.Models](#)

Assembly: BOOSE_WebApplication.dll

```
public class ErrorViewModel
```

Inheritance

[object](#) ← ErrorViewModel

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Properties

RequestId

```
public string? RequestId { get; set; }
```

Property Value

[string](#)

ShowRequestId

```
public bool ShowRequestId { get; }
```

Property Value

[bool](#)

Namespace ase_assessment

Classes

[AppCanvas](#)

Represents a canvas for drawing shapes and lines with a customizable pen color. Implements the Icanvas interface.

[AppCircle](#)

Represents the command to draw the Circle on the canvas.

[AppCommandFactory](#)

Implements AppCommandFactory class for creating commands.

[AppPolygon](#)

Represents a command to draw a polygon on the canvas. Inherits from BOOSE.Evaluation class.

[AppRectangle](#)

Represents the command to draw a Rectangle on the canvas.

[AppTriangle](#)

Represents a command to draw a triangle on the canvas.

[AppWrite](#)

Represents write command This is the write class which implements Evaluation class to perform command execution and parameter checking.

[ArrayTest](#)

Represents testing for restricted and unrestricted Array classes.

[CustomMethod](#)

Represents the customMethod that inherits from the parent class Method.

[CustomParser](#)

Represents a custom parser that extends the functionality of the base Parser class. Allows comments when parsing commands adding additional logic.

[CustomStoredProgram](#)

Represents a custom stored program that extends the functionality of the base StoredProgram class. Provides additional functionality such as accessing private fields using Reflecton.

[ForTest](#)

Represents testing for restricted and unrestricted For classes.

[Form1](#)

This is the main form which handles GUI and interaction between user and program

IfTest

Represents testing for restricted and unrestricted If classes.

IntTest

Represents testing for restricted and unrestricted Int classes.

MethodTest

Represents testing for restricted and unrestricted Int classes.

RealTest

Represents testing for restricted and unrestricted Real classes.

UnrestrictedArray

Represents the array class that inherits the base class called Array.

UnrestrictedElse

Represents the unrestricted version of the else class with reduced restrictions.

UnrestrictedEnd

Represents the unrestricted version of the End class with reduced restrictions.

UnrestrictedFor

Represents the unrestricted version of the For class. Allows the custom behaviour overriding Restrictions method.

UnrestrictedIf

Represents the unrestricted version of the If class with reduced restrictions.

UnrestrictedInt

Represents the child class which inherits parent class Int.

UnrestrictedReal

Represents the real class that inherits parent class. Provides the implementation of Execution and Restrictions.

UnrestrictedWhile

Represents the Unrestricted version of the While class. Additionally, it contains reduced restrictions for more flexibility.

WhileTest

Class AppCanvas

Namespace: [ase assessment](#)

Assembly: ase_assessment.dll

Represents a canvas for drawing shapes and lines with a customizable pen color. Implements the Icanvas interface.

```
public class AppCanvas : ICanvas
```

Inheritance

[object](#) ← AppCanvas

Implements

ICanvas

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Constructors

AppCanvas()

Intializes a new instance of the class AppCanvas

```
public AppCanvas()
```

Properties

PenColour

Gets or sets the Color of the pen.

```
public object PenColour { get; set; }
```

Property Value

[object](#)

Xpos

Gets or sets the X position of the pen on the canvas.

```
public int Xpos { get; set; }
```

Property Value

[int](#)

Ypos

Gets or sets the Y position of the pen on the canvas.

```
public int Ypos { get; set; }
```

Property Value

[int](#)

Methods

Circle(int, bool)

Draws a circle on the canvas.

```
public void Circle(int radius, bool filled)
```

Parameters

radius [int](#)

The radius of the circle.

filled `bool`

Indicates whether the circle should be filled

Exceptions

CanvasException

Exception is thrown when the radius is negative.

`Clear()`

Clears the canvas and resets to default color Gray.

```
public void Clear()
```

`DrawTo(int, int)`

Draws to the X and Y position from current pen position.

```
public void DrawTo(int toX, int toY)
```

Parameters

`toX` `int`

X coordinates of the end point.

`toY` `int`

Y coordinates of the end point.

Exceptions

CanvasException

Throws error when pen size is less than 0 and greater than canvas size.

MoveTo(int, int)

Moves from initial position to X, Y coordinates.

```
public void MoveTo(int x, int y)
```

Parameters

x [int](#)

X coordinates to move pen to.

y [int](#)

Y coordinates to move pen to.

Exceptions

CanvasException

Throws error when pen position is outside the canvas.

Polygon(List<Point>)

Draw a polygon based on a list of points.

```
public void Polygon(List<Point> points)
```

Parameters

points [List](#)<[Point](#)>

Rect(int, int, bool)

Draws rectangle.

```
public void Rect(int width, int height, bool filled)
```

Parameters

width [int](#)

Width of the rectangle.

height [int](#)

Height of the rectangle.

filled [bool](#)

Denotes either rectangle is filled or not.

Exceptions

CanvasException

Reset()

Resets the canvas to initial state.

```
public void Reset()
```

Set(int, int)

Sets the canvas size.

```
public void Set(int xsize, int ysize)
```

Parameters

xsize [int](#)

New width of the canvas.

ysize [int](#)

New weight of the canvas.

SetColour(int, int, int)

Sets the pen color with the use of RGB values.

```
public void SetColour(int red, int green, int blue)
```

Parameters

red [int](#)

green [int](#)

blue [int](#)

Exceptions

CanvasException

Tri(int, int)

Draws a triangle with specified height and weight in the current position.

```
public void Tri(int width, int height)
```

Parameters

width [int](#)

Width of the triangle

height [int](#)

Height of the triangle

Exceptions

CanvasException

Throws error when

WriteText(string)

Writes text in the canvas.

```
public void WriteText(string text)
```

Parameters

text [string](#)

Text that to be written on the canvas.

getBitmap()

Fetch the current bitmap which reprents the canvas.

```
public object getBitmap()
```

Returns

[object](#)

Bitmap image of the canvas.

Class AppCircle

Namespace: [ase assessment](#)

Assembly: ase_assessment.dll

Represents the command to draw the Circle on the canvas.

```
public class AppCircle : CommandTwoParameters, ICommand
```

Inheritance

[object](#) ← Command ← CanvasCommand ← CommandOneParameter ← CommandTwoParameters ← AppCircle

Implements

ICommand

Inherited Members

CommandTwoParameters.param2 , CommandTwoParameters.param2unprocessed ,
CommandOneParameter.param1 , CommandOneParameter.param1unprocessed ,
CanvasCommand.yPos , CanvasCommand.xPos , CanvasCommand.canvas , CanvasCommand.Canvas ,
Command.program , Command.parameterList , Command.parameters , Command.paramsint ,
[Command.Set\(StoredProgram, string\)](#) , Command.Compile() , [Command.ProcessParameters\(string\)](#) ,
Command.ToString() , Command.Program , Command.Name , Command.ParameterList ,
Command.Parameters , Command.Paramsint , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) ,
[object.ReferenceEquals\(object, object\)](#)

Constructors

AppCircle()

Default constructor.

```
public AppCircle()
```

AppCircle(Canvas, int, bool)

Constructor with parameters.

```
public AppCircle(Canvas c, int radius, bool filled)
```

Parameters

c Canvas

The canvas where circle is drawn.

radius [int](#)

The radius of the circle.

filled [bool](#)

If the filled is true then filled circle will be drawn on the canvas.

Methods

CheckParameters(string[])

Validates the parameters of the circle.

```
public override void CheckParameters(string[] parameterList)
```

Parameters

parameterList [string](#)[]

An array of parameters such as radius and filled.

Exceptions

CommandException

An exception will be thrown if invalid parameter are provided.

Execute()

Executes the circle command. Draws circle on the canvas.

```
public override void Execute()
```

Class AppCommandFactory

Namespace: [ase assessment](#)

Assembly: ase_assessment.dll

Implements AppCommandFactory class for creating commands.

```
public class AppCommandFactory : CommandFactory, ICommandFactory
```

Inheritance

[object](#) ← CommandFactory ← AppCommandFactory

Implements

ICommandFactory

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Constructors

AppCommandFactory()

```
public AppCommandFactory()
```

AppCommandFactory(AppCanvas)

Constructor that requires an instance of an app canvas to be passed.

```
public AppCommandFactory(AppCanvas canvas)
```

Parameters

canvas [AppCanvas](#)

Methods

MakeCommand(string)

Takes the command according to the user input.

```
public override ICommand MakeCommand(string commandType)
```

Parameters

commandType [string](#)

The type of command.

Returns

ICommand

Class AppPolygon

Namespace: [ase assessment](#)

Assembly: ase_assessment.dll

Represents a command to draw a polygon on the canvas. Inherits from BOOSE.Evaluation class.

```
public class AppPolygon : Evaluation, ICommand
```

Inheritance

[object](#) ← Command ← Evaluation ← AppPolygon

Implements

ICommand

Inherited Members

Evaluation.expression , Evaluation.evaluatedExpression , Evaluation.varName , Evaluation.value ,
[Evaluation.CheckParameters\(string\[\]\)](#) , [Evaluation.ProcessExpression\(string\)](#) , Evaluation.Expression ,
Evaluation.VarName , Evaluation.Value , Evaluation.Local , Command.program , Command.parameterList ,
Command.parameters , Command.paramsint , [Command.Set\(StoredProgram, string\)](#) ,
[Command.ProcessParameters\(string\)](#) , Command.ToString() , Command.Program , Command.Name ,
Command.ParameterList , Command.Parameters , Command.Paramsint , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#)

Constructors

AppPolygon(AppCanvas)

Initializes a new instance of the [AppPolygon](#) class.

```
public AppPolygon(AppCanvas canvas)
```

Parameters

canvas [AppCanvas](#)

The canvas to draw on.

Methods

Compile()

Compiles the Polygon command by parsing the parameter list and creating the points.

```
public override void Compile()
```

Exceptions

CommandException

Thrown when the parameters are invalid.

Execute()

Executes the polygon command and draws the polygon using the AppCanvas.

```
public override void Execute()
```

GetPoints()

Gets the points of the polygon.

```
public List<Point> GetPoints()
```

Returns

[List](#)<[Point](#)>

The list of points that the polygon consists of.

Class AppRectangle

Namespace: [ase assessment](#)

Assembly: ase_assessment.dll

Represents the command to draw a Rectangle on the canvas.

```
public class AppRectangle : CommandOneParameter, ICommand
```

Inheritance

[object](#) ← Command ← CanvasCommand ← CommandOneParameter ← AppRectangle

Implements

ICommand

Inherited Members

CommandOneParameter.param1 , CommandOneParameter.param1unprocessed ,
CanvasCommand.yPos , CanvasCommand.xPos , CanvasCommand.canvas , CanvasCommand.Canvas ,
Command.program , Command.parameterList , Command.parameters , Command.paramsint ,
[Command.Set\(StoredProgram, string\)](#) , Command.Compile() , [Command.ProcessParameters\(string\)](#) ,
Command.ToString() , Command.Program , Command.Name , Command.ParameterList ,
Command.Parameters , Command.Paramsint , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) ,
[object.ReferenceEquals\(object, object\)](#)

Constructors

AppRectangle()

Default constructor.

```
public AppRectangle()
```

AppRectangle(Canvas, int, int, bool)

Constructor with parameters.

```
public AppRectangle(Canvas c, int width, int height, bool filled)
```

Parameters

c Canvas

The canvas where Rectangle will be drawn.

width int ↴

The width of the Rectangle.

height int ↴

The height of the Rectangle.

filled bool ↴

If the filled is true then filled rectangle will be drawn on the canvas.

Methods

CheckParameters(string[])

Method to check the valid parameter of the Rectangle.

```
public override void CheckParameters(string[] parameterList)
```

Parameters

parameterList string ↴[]

An Array of string which represents parameters i.e. width, height, filled

Exceptions

CommandException

Throws exception if parameter is null and not valid.

Execute()

Executes the rectangle command. Draws the rectangle in the canvas with provided width, height, filled.

```
public override void Execute()
```

Class AppTriangle

Namespace: [ase assessment](#)

Assembly: ase_assessment.dll

Represents a command to draw a triangle on the canvas.

```
public class AppTriangle : CommandTwoParameters, ICommand
```

Inheritance

[object](#) ← Command ← CanvasCommand ← CommandOneParameter ← CommandTwoParameters ← AppTriangle

Implements

ICommand

Inherited Members

CommandTwoParameters.param2 , CommandTwoParameters.param2unprocessed ,
CommandOneParameter.param1 , CommandOneParameter.param1unprocessed ,
CanvasCommand.yPos , CanvasCommand.xPos , CanvasCommand.canvas , CanvasCommand.Canvas ,
Command.program , Command.parameterList , Command.parameters , Command.paramsint ,
[Command.Set\(StoredProgram, string\)](#) , Command.Compile() , [Command.ProcessParameters\(string\)](#) ,
Command.ToString() , Command.Program , Command.Name , Command.ParameterList ,
Command.Parameters , Command.Paramsint , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) ,
[object.ReferenceEquals\(object, object\)](#)

Constructors

AppTriangle()

Default constructor

```
public AppTriangle()
```

AppTriangle(Canvas, int, int)

Constructor with parameters

```
public AppTriangle(Canvas c, int width, int height)
```

Parameters

c Canvas

Canvas on which triangle will be drawn

width int

The width of the triangle.

height int

The height of the triangle.

Methods

CheckParameters(string[])

Method to check the valid parameters of triangle.

```
public override void CheckParameters(string[] parameterList)
```

Parameters

parameterList string[]

An array of string parameters i.e width and height.

Exceptions

CommandException

Throws an error if parameter is not provided and if either parameter is not valid integer.

Execute()

Executes the triangle command

```
public override void Execute()
```

Class AppWrite

Namespace: [ase_assessment](#)

Assembly: ase_assessment.dll

Represents write command This is the write class which implements Evaluation class to perform command execution and parameter checking.

```
public class AppWrite : Evaluation, ICommand
```

Inheritance

[object](#) ← Command ← Evaluation ← AppWrite

Implements

ICommand

Inherited Members

Evaluation.expression , Evaluation.evaluatedExpression , Evaluation.varName , Evaluation.value ,
Evaluation.Compile() , [Evaluation.ProcessExpression\(string\)](#) , Evaluation.Expression ,
Evaluation.VarName , Evaluation.Value , Evaluation.Local , Command.program , Command.parameterList ,
Command.parameters , Command.paramsint , [Command.Set\(StoredProgram, string\)](#) ,
[Command.ProcessParameters\(string\)](#) , Command.ToString() , Command.Program , Command.Name ,
Command.ParameterList , Command.Parameters , Command.Paramsint , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#)

Properties

OutputCallback

Static output callback handler (a delegate) for the output. this delegate is called evaluated expression if it is not null.

```
public static Action<string>? OutputCallback { get; set; }
```

Property Value

[Action](#) <[string](#)>

Methods

CheckParameters(string[])

Checks if the parameters passed to the 'Write' command are verified. Guarantees atleast one parameter is passes to the write command.

```
public override void CheckParameters(string[] parameter)
```

Parameters

parameter string[]

The parameter which needs to be checked.

Exceptions

CommandException

Throws error when parameter length is less than 1.

Execute()

Executes the 'Write' command by evaluating the first parameters. if not null then passess to the output callback.

```
public override void Execute()
```

Exceptions

CommandException

Throws error when parameter is null.

Class ArrayTest

Namespace: [ase assessment](#)

Assembly: unit_testing.dll

Represents testing for restricted and unrestricted Array classes.

```
[TestClass]  
public class ArrayTest
```

Inheritance

[object](#) ← ArrayTest

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Methods

NoRestrictionTest()

Test the unrestricted Array class. Ensures that multiple objects can be created.

```
[TestMethod]  
public void NoRestrictionTest()
```

Class CustomMethod

Namespace: [ase assessment](#)

Assembly: ase_assessment.dll

Represents the customMethod that inherits from the parent clas Method.

```
public class CustomMethod : Method, ICommand
```

Inheritance

[object](#) ← Command ← Evaluation ← Boolean ← ConditionalCommand ← CompoundCommand ← Method ← CustomMethod

Implements

ICommand

Inherited Members

[Method.CheckParameters\(string\[\]\)](#) , Method.LocalVariables , Method.MethodName , Method.Type ,
CompoundCommand.ReduceRestrictions() , CompoundCommand.CorrectingCommand ,
ConditionalCommand.endLineNumber , ConditionalCommand.EndLineNumber ,
ConditionalCommand.Condition , ConditionalCommand.LineNumber , ConditionalCommand.CondType ,
ConditionalCommand.ReturnLineNumber , Boolean.BoolValue , Evaluation.expression ,
Evaluation.evaluatedExpression , Evaluation.varName , Evaluation.value ,
[Evaluation.ProcessExpression\(string\)](#) , Evaluation.Expression , Evaluation.VarName , Evaluation.Value ,
Evaluation.Local , Command.program , Command.parameterList , Command.parameters ,
Command.paramsint , [Command.Set\(StoredProgram, string\)](#) , [Command.ProcessParameters\(string\)](#) ,
Command.ToString() , Command.Program , Command.Name , Command.ParameterList ,
Command.Parameters , Command.Paramsint , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) ,
[object.ReferenceEquals\(object, object\)](#)

Constructors

CustomMethod()

Default constructor for the CustomMethod class.

```
public CustomMethod()
```

Methods

Compile()

Represents the overriden compile method. It calls the base class's compile method first.

```
public override void Compile()
```

Execute()

Represents Execute method that runs the custom method. Adds the method to the program.

```
public override void Execute()
```

Restrictions()

Implements the custom behaviour by overriding Restrictions method.

```
public override void Restrictions()
```

Class CustomParser

Namespace: [ase assessment](#)

Assembly: ase_assessment.dll

Represents a custom parser that extends the functionality of the base Parser class. Allows comments when parsing commands adding additional logic.

```
public class CustomParser : Parser, IParser
```

Inheritance

[object](#) ← Parser ← CustomParser

Implements

IParser

Inherited Members

[Parser.ParseProgram\(string\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Constructors

CustomParser(CommandFactory, StoredProgram)

Initializes the new instances of the [CustomParser](#) class.

```
public CustomParser(CommandFactory factory, StoredProgram program)
```

Parameters

factory CommandFactory

The CommandFactory used to create commands.

program StoredProgram

The StoredProgram is associated with this Parser.

Methods

ParseCommand(string)

Parses a command from a given line of input.

```
public override ICommand ParseCommand(string line)
```

Parameters

line string ↗

The input line to parse.

Returns

ICommand

Returns ICommand object representing the parsed command. Returns null if the command is comment.

Class CustomeStoredProgram

Namespace: [ase assessment](#)

Assembly: ase_assessment.dll

Represents a custom stored program that extends the functionality of the base StoredProgram class. Provides additional functionality such as accessing private fields using Reflecton.

```
public class CustomeStoredProgram : StoredProgram, IList, ICollection, IEnumerable,  
ICloneable, IStoredProgram
```

Inheritance

[object](#) ← [ArrayList](#) ← [StoredProgram](#) ← [CustomeStoredProgram](#)

Implements

[IList](#), [ICollection](#), [IEnumerable](#), [ICloneable](#), [IStoredProgram](#)

Inherited Members

[StoredProgram.SyntaxOk](#), [StoredProgram.AddMethod\(Method\)](#), [StoredProgram.GetMethod\(string\)](#),
[StoredProgram.AddVariable\(Evaluation\)](#), [StoredProgram.GetVariable\(string\)](#),
[StoredProgram.GetVariable\(int\)](#), [StoredProgram.FindVariable\(Evaluation\)](#),
[StoredProgram.FindVariable\(string\)](#), [StoredProgram.VariableExists\(string\)](#),
[StoredProgram.GetVarValue\(string\)](#), [StoredProgram.UpdateVariable\(string, int\)](#),
[StoredProgram.UpdateVariable\(string, double\)](#), [StoredProgram.UpdateVariable\(string, bool\)](#),
[StoredProgram.DeleteVariable\(string\)](#), [StoredProgram.IsExpression\(string\)](#),
[StoredProgram.EvaluateExpressionWithString\(string\)](#), [StoredProgram.EvaluateExpression\(string\)](#),
[StoredProgram.Push\(ConditionalCommand\)](#), [StoredProgram.Pop\(\)](#), [StoredProgram.Add\(Command\)](#),
[StoredProgram.NextCommand\(\)](#), [StoredProgram.ResetProgram\(\)](#), [StoredProgram.Commandsleft\(\)](#),
[StoredProgram.PC](#), [ArrayList.Adapter\(IList\)](#), [ArrayList.Add\(object\)](#),
[ArrayList.AddRange\(Collection\)](#), [ArrayList.BinarySearch\(int, int, object, Comparer\)](#),
[ArrayList.BinarySearch\(object\)](#), [ArrayList.BinarySearch\(object, Comparer\)](#), [ArrayList.Clear\(\)](#),
[ArrayList.Clone\(\)](#), [ArrayList.Contains\(object\)](#), [ArrayList.CopyTo\(Array\)](#),
[ArrayList.CopyTo\(Array, int\)](#), [ArrayList.CopyTo\(int, Array, int, int\)](#), [ArrayList.FixedSize\(ArrayList\)](#),
[ArrayList.FixedSize\(IList\)](#), [ArrayList.GetEnumerator\(\)](#), [ArrayList.GetEnumerator\(int, int\)](#),
[ArrayList.GetRange\(int, int\)](#), [ArrayList.IndexOf\(object\)](#), [ArrayList.IndexOf\(object, int\)](#),
[ArrayList.IndexOf\(object, int, int\)](#), [ArrayList.Insert\(int, object\)](#),
[ArrayList.InsertRange\(int, Collection\)](#), [ArrayList.LastIndexOf\(object\)](#),
[ArrayList.LastIndexOf\(object, int\)](#), [ArrayList.LastIndexOf\(object, int, int\)](#),
[ArrayList.ReadOnly\(ArrayList\)](#), [ArrayList.ReadOnly\(IList\)](#), [ArrayList.Remove\(object\)](#),
[ArrayList.RemoveAt\(int\)](#), [ArrayList.RemoveRange\(int, int\)](#), [ArrayList.Repeat\(object, int\)](#),

[ArrayList.Reverse\(\)](#) , [ArrayList.Reverse\(int, int\)](#) , [ArrayList.SetRange\(int, ICollection\)](#) ,
[ArrayList.Sort\(\)](#) , [ArrayList.Sort\(IComparer\)](#) , [ArrayList.Sort\(int, int, IComparer\)](#) ,
[ArrayList.Synchronized\(ArrayList\)](#) , [ArrayList.Synchronized\(IList\)](#) , [ArrayList.ToArray\(\)](#) ,
[ArrayList.ToArray\(Type\)](#) , [ArrayList.TrimToSize\(\)](#) , [ArrayList.Capacity](#) , [ArrayList.Count](#) ,
[ArrayList.IsFixedSize](#) , [ArrayList.IsReadOnly](#) , [ArrayList.IsSynchronized](#) , [ArrayList.this\[int\]](#) ,
[ArrayList.SyncRoot](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,
[object.ToString\(\)](#)

Constructors

CustomeStoredProgram(ICanvas)

```
public CustomeStoredProgram(ICanvas canvas)
```

Parameters

canvas ICanvas

Methods

Run()

Executes the stored program by processing commands sequentially.

```
public override void Run()
```

Exceptions

StoredProgramException

If invalid syntax or execution defined constraints then throw exception.

Class ForTest

Namespace: [ase assessment](#)

Assembly: unit_testing.dll

Represents testing for restricted and unrestricted For classes.

```
[TestClass]  
public class ForTest
```

Inheritance

[object](#) ← ForTest

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Methods

NoRestrictionTest()

Test the unrestricted For class. Ensures that multiple objects can be created.

```
[TestMethod]  
public void NoRestrictionTest()
```

RestrictionTest()

Tests the restriction of the For Class.

```
[TestMethod]  
public void RestrictionTest()
```

Class Form1

Namespace: [ase assessment](#)

Assembly: ase_assessment.dll

This is the main form which handles GUI and interaction between user and program

```
public class Form1 : Form, IDropTarget, ISynchronizeInvoke, IWin32Window,  
IBindableComponent, IComponent, IDisposable, IContainerControl
```

Inheritance

```
object ↳ ← MarshalByRefObject ↳ ← Component ↳ ← Control ↳ ← ScrollableControl ↳ ←  
ContainerControl ↳ ← Form ↳ ← Form1
```

Implements

```
IDropTarget ↳ , ISynchronizeInvoke ↳ , IWin32Window ↳ , IBindableComponent ↳ , IComponent ↳ ,  
IDisposable ↳ , IContainerControl ↳
```

Inherited Members

```
Form.SetVisibleCore(bool) ↳ , Form.Activate() ↳ , Form.ActivateMdiChild(Form) ↳ ,  
Form.AddOwnedForm(Form) ↳ , Form.AdjustFormScrollbars(bool) ↳ , Form.Close() ↳ ,  
Form.CreateAccessibilityInstance() ↳ , Form.CreateControlsInstance() ↳ , Form.CreateHandle() ↳ ,  
Form.DefWndProc(ref Message) ↳ , Form.ProcessMnemonic(char) ↳ , Form.CenterToParent() ↳ ,  
Form.CenterToScreen() ↳ , Form.LayoutMdi(MdiLayout) ↳ , Form.OnActivated(EventArgs) ↳ ,  
Form.OnBackgroundImageChanged(EventArgs) ↳ ,  
Form.OnBackgroundImageLayoutChanged(EventArgs) ↳ , Form.OnClosing(CancelEventArgs) ↳ ,  
Form.OnClosed(EventArgs) ↳ , Form.OnFormClosing(FormClosingEventArgs) ↳ ,  
Form.OnFormClosed(FormClosedEventArgs) ↳ , Form.OnCreateControl() ↳ ,  
Form.OnDeactivate(EventArgs) ↳ , Form.OnEnabledChanged(EventArgs) ↳ , Form.OnEnter(EventArgs) ↳ ,  
Form.OnFontChanged(EventArgs) ↳ , Form.OnGotFocus(EventArgs) ↳ ,  
Form.OnHandleCreated(EventArgs) ↳ , Form.OnHandleDestroyed(EventArgs) ↳ ,  
Form.OnHelpButtonClicked(CancelEventArgs) ↳ , Form.OnLayout(LayoutEventArgs) ↳ ,  
Form.OnLoad(EventArgs) ↳ , Form.OnMaximizedBoundsChanged(EventArgs) ↳ ,  
Form.OnMaximumSizeChanged(EventArgs) ↳ , Form.OnMinimumSizeChanged(EventArgs) ↳ ,  
Form.OnInputLanguageChanged(InputLanguageChangedEventArgs) ↳ ,  
Form.OnInputLanguageChanging(InputLanguageChangingEventArgs) ↳ ,  
Form.OnVisibleChanged(EventArgs) ↳ , Form.OnMdiChildActivate(EventArgs) ↳ ,  
Form.OnMenuStart(EventArgs) ↳ , Form.OnMenuComplete(EventArgs) ↳ ,  
Form.OnPaint(PaintEventArgs) ↳ , Form.OnResize(EventArgs) ↳ ,  
Form.OnDpiChanged(DpiChangedEventArgs) ↳ , Form.OnGetDpiScaledSize(int, int, ref Size) ↳ ,
```

[Form.OnRightToLeftLayoutChanged\(EventArgs\)](#) , [Form.OnShown\(EventArgs\)](#) ,
[Form.OnTextChanged\(EventArgs\)](#) , [Form.ProcessCmdKey\(ref Message, Keys\)](#) ,
[Form.ProcessDialogKey\(Keys\)](#) , [Form.ProcessDialogChar\(char\)](#) ,
[Form.ProcessKeyPreview\(ref Message\)](#) , [Form.ProcessTabKey\(bool\)](#) ,
[Form.RemoveOwnedForm\(Form\)](#) , [Form.Select\(bool, bool\)](#) ,
[Form.ScaleMinAxisSize\(float, float, bool\)](#) ,
[Form.GetScaledBounds\(Rectangle, SizeF, BoundsSpecified\)](#) ,
[Form.ScaleControl\(SizeF, BoundsSpecified\)](#) , [Form.SetBoundsCore\(int, int, int, int, BoundsSpecified\)](#) ,
[Form.SetClientSizeCore\(int, int\)](#) , [Form.SetDesktopBounds\(int, int, int, int\)](#) ,
[Form.SetDesktopLocation\(int, int\)](#) , [Form.Show\(IWin32Window\)](#) , [Form.ShowDialog\(\)](#) ,
[Form.ShowDialog\(IWin32Window\)](#) , [Form.ToString\(\)](#) , [Form.UpdateDefaultButton\(\)](#) ,
[Form.OnResizeBegin\(EventArgs\)](#) , [Form.OnResizeEnd\(EventArgs\)](#) ,
[Form.OnStyleChanged\(EventArgs\)](#) , [Form.ValidateChildren\(\)](#) ,
[Form.ValidateChildren\(ValidationConstraints\)](#) , [Form.WndProc\(ref Message\)](#) , [Form.AcceptButton](#) ,
[Form.ActiveForm](#) , [Form.ActiveMdiChild](#) , [Form.AllowTransparency](#) , [Form.AutoScroll](#) ,
[Form.AutoSize](#) , [Form.AutoSizeMode](#) , [Form.AutoValidate](#) , [Form.BackColor](#) ,
[Form.FormBorderStyle](#) , [Form.CancelButton](#) , [Form.ClientSize](#) , [Form.ControlBox](#) ,
[Form.CreateParams](#) , [Form.DefaultImeMode](#) , [Form.DefaultSize](#) , [Form.DesktopBounds](#) ,
[Form/DesktopLocation](#) , [Form/DialogResult](#) , [Form/HelpButton](#) , [Form/Icon](#) , [Form/IsMdiChild](#) ,
[Form/IsMdiContainer](#) , [Form/IsRestrictedWindow](#) , [Form/KeyPreview](#) , [Form/Location](#) ,
[Form/MaximizedBounds](#) , [Form/MaximumSize](#) , [Form/MainMenuStrip](#) , [Form/MinimumSize](#) ,
[Form/MaximizeBox](#) , [Form/MdiChildren](#) , [Form/MdiChildrenMinimizedAnchorBottom](#) ,
[Form/MdiParent](#) , [Form/MinimizeBox](#) , [Form/Modal](#) , [Form/Opacity](#) , [Form/OwnedForms](#) ,
[Form/Owner](#) , [Form/RestoreBounds](#) , [Form/RightToLeftLayout](#) , [Form>ShowInTaskbar](#) ,
[Form>ShowIcon](#) , [Form>ShowWithoutActivation](#) , [Form/Size](#) , [Form/SizeGripStyle](#) ,
[Form/StartPosition](#) , [Form/Text](#) , [Form/TopLevel](#) , [Form/TopMost](#) , [Form/TransparencyKey](#) ,
[Form/WindowState](#) , [Form/AutoSizeChanged](#) , [Form/AutoValidateChanged](#) ,
[Form/HelpButtonClicked](#) , [Form/MaximizedBoundsChanged](#) , [Form/MaximumSizeChanged](#) ,
[Form/MinimumSizeChanged](#) , [Form/Activated](#) , [Form/Deactivate](#) , [Form/FormClosing](#) ,
[Form/FormClosed](#) , [Form/Load](#) , [Form/MdiChildActivate](#) , [Form/MenuComplete](#) ,
[Form/MenuStart](#) , [Form/InputLanguageChanged](#) , [Form/InputLanguageChanging](#) ,
[Form/RightToLeftLayoutChanged](#) , [Form/Shown](#) , [Form/DpiChanged](#) , [Form/ResizeBegin](#) ,
[Form/ResizeEnd](#) , [ContainerControl.OnAutoValidateChanged\(EventArgs\)](#) ,
[ContainerControl.OnMove\(EventArgs\)](#) , [ContainerControl.OnParentChanged\(EventArgs\)](#) ,
[ContainerControl.PerformLayout\(\)](#) , [ContainerControl.RescaleConstantsForDpi\(int, int\)](#) ,
[ContainerControl.Validate\(\)](#) , [ContainerControl.Validate\(bool\)](#) ,
[ContainerControl.AutoScaleDimensions](#) , [ContainerControl.AutoScaleFactor](#) ,
[ContainerControl.AutoScaleMode](#) , [ContainerControl.BindingContext](#) ,
[ContainerControl.CanEnableIme](#) , [ContainerControl.ActiveControl](#) ,
[ContainerControl.CurrentAutoScaleDimensions](#) , [ContainerControl.ParentForm](#) ,

[ScrollableControl.ScrollStateAutoScrolling](#) , [ScrollableControl.ScrollStateHScrollVisible](#) ,
[ScrollableControl.ScrollStateVScrollVisible](#) , [ScrollableControl.ScrollStateUserHasScrolled](#) ,
[ScrollableControl.ScrollStateFullDrag](#) , [ScrollableControl.GetScrollState\(int\)](#) ,
[ScrollableControl.OnMouseWheel\(MouseEventArgs\)](#) ,
[ScrollableControl.OnRightToLeftChanged\(EventArgs\)](#) ,
[ScrollableControl.OnPaintBackground\(PaintEventArgs\)](#) ,
[ScrollableControl.OnPaddingChanged\(EventArgs\)](#) , [ScrollableControl.SetDisplayRectLocation\(int, int\)](#) ,
[ScrollableControl.ScrollControlIntoView\(Control\)](#) , [ScrollableControl.ScrollToControl\(Control\)](#) ,
[ScrollableControl.OnScroll\(ScrollEventArgs\)](#) , [ScrollableControl.SetAutoScrollMargin\(int, int\)](#) ,
[ScrollableControl.SetScrollState\(int, bool\)](#) , [ScrollableControl.AutoScrollMargin](#) ,
[ScrollableControl.AutoScrollPosition](#) , [ScrollableControl.AutoScrollMinSize](#) ,
[ScrollableControl.DisplayRectangle](#) , [ScrollableControl.HScroll](#) , [ScrollableControl.HorizontalScroll](#) ,
[ScrollableControl.VScroll](#) , [ScrollableControl.VerticalScroll](#) , [ScrollableControl.Scroll](#) ,
[Control.GetAccessibilityObjectById\(int\)](#) , [Control.SetAutoSizeMode\(AutoSizeMode\)](#) ,
[Control.GetAutoSizeMode\(\)](#) , [Control.GetPreferredSize\(Size\)](#) ,
[Control.AccessibilityNotifyClients\(AccessibleEvents, int\)](#) ,
[Control.AccessibilityNotifyClients\(AccessibleEvents, int, int\)](#) , [Control.BeginInvoke\(Delegate\)](#) ,
[Control.BeginInvoke\(Action\)](#) , [Control.BeginInvoke\(Delegate, params object\[\]\)](#) ,
[Control.BringToFront\(\)](#) , [Control.Contains\(Control\)](#) , [Control.CreateGraphics\(\)](#) ,
[Control.CreateControl\(\)](#) , [Control.DestroyHandle\(\)](#) , [Control.DoDragDrop\(object, DragDropEffects\)](#) ,
[Control.DoDragDrop\(object, DragDropEffects, Bitmap, Point, bool\)](#) ,
[Control.DrawToBitmap\(Bitmap, Rectangle\)](#) , [Control.EndInvoke\(IAsyncResult\)](#) , [Control.FindForm\(\)](#) ,
[Control.GetTopLevel\(\)](#) , [Control.RaiseKeyEvent\(object, KeyEventArgs\)](#) ,
[Control.RaiseMouseEvent\(object, MouseEventArgs\)](#) , [Control.Focus\(\)](#) ,
[Control.FromChildHandle\(nint\)](#) , [Control.FromHandle\(nint\)](#) ,
[Control.GetChildAtPoint\(Point, GetChildAtPointSkip\)](#) , [Control.GetChildAtPoint\(Point\)](#) ,
[Control.GetContainerControl\(\)](#) , [Control.GetNextControl\(Control, bool\)](#) ,
[Control.GetStyle\(ControlStyles\)](#) , [Control.Hide\(\)](#) , [Control.InitLayout\(\)](#) , [Control.Invalidate\(Region\)](#) ,
[Control.Invalidate\(Region, bool\)](#) , [Control.Invalidate\(\)](#) , [Control.Invalidate\(bool\)](#) ,
[Control.Invalidate\(Rectangle\)](#) , [Control.Invalidate\(Rectangle, bool\)](#) , [Control.Invoke\(Action\)](#) ,
[Control.Invoke\(Delegate\)](#) , [Control.Invoke\(Delegate, params object\[\]\)](#) ,
[Control.Invoke<T>\(Func<T>\)](#) , [Control.InvokePaint\(Control, PaintEventArgs\)](#) ,
[Control.InvokePaintBackground\(Control, PaintEventArgs\)](#) , [Control.IsKeyLocked\(Keys\)](#) ,
[Control.IsInputChar\(char\)](#) , [Control.IsInputKey\(Keys\)](#) , [Control.IsMnemonic\(char, string\)](#) ,
[Control.LogicalToDeviceUnits\(int\)](#) , [Control.LogicalToDeviceUnits\(Size\)](#) ,
[Control.ScaleBitmapLogicalToDevice\(ref Bitmap\)](#) , [Control.NotifyInvalidate\(Rectangle\)](#) ,
[Control.InvokeOnClick\(Control, EventArgs\)](#) , [Control.OnAutoSizeChanged\(EventArgs\)](#) ,
[Control.OnBackColorChanged\(EventArgs\)](#) , [Control.OnBindingContextChanged\(EventArgs\)](#) ,
[Control.OnCausesValidationChanged\(EventArgs\)](#) , [Control.OnContextMenuStripChanged\(EventArgs\)](#) ,
[Control.OnCursorChanged\(EventArgs\)](#) , [Control.OnDataContextChanged\(EventArgs\)](#) ,

[Control.OnDockChanged\(EventArgs\)](#) , [Control.OnForeColorChanged\(EventArgs\)](#) ,
[Control.OnNotifyMessage\(Message\)](#) , [Control.OnParentBackColorChanged\(EventArgs\)](#) ,
[Control.OnParentBackgroundImageChanged\(EventArgs\)](#) ,
[Control.OnParentBindingContextChanged\(EventArgs\)](#) , [Control.OnParentCursorChanged\(EventArgs\)](#) ,
[Control.OnParentDataContextChanged\(EventArgs\)](#) , [Control.OnParentEnabledChanged\(EventArgs\)](#) ,
[Control.OnParentFontChanged\(EventArgs\)](#) , [Control.OnParentForeColorChanged\(EventArgs\)](#) ,
[Control.OnParentRightToLeftChanged\(EventArgs\)](#) , [Control.OnParentVisibleChanged\(EventArgs\)](#) ,
[Control.OnPrint\(PaintEventArgs\)](#) , [Control.OnTabIndexChanged\(EventArgs\)](#) ,
[Control.OnTabStopChanged\(EventArgs\)](#) , [Control.OnClick\(EventArgs\)](#) ,
[Control.OnClientSizeChanged\(EventArgs\)](#) , [Control.OnControlAdded\(ControlEventArgs\)](#) ,
[Control.OnControlRemoved\(ControlEventArgs\)](#) , [Control.OnLocationChanged\(EventArgs\)](#) ,
[Control.OnDoubleClick\(EventArgs\)](#) , [Control.OnDragEnter\(DragEventArgs\)](#) ,
[Control.OnDragOver\(DragEventArgs\)](#) , [Control.OnDragLeave\(EventArgs\)](#) ,
[Control.OnDragDrop\(DragEventArgs\)](#) , [Control.OnGiveFeedback\(GiveFeedbackEventArgs\)](#) ,
[Control.InvokeGotFocus\(Control, EventArgs\)](#) , [Control.OnHelpRequested\(HelpEventArgs\)](#) ,
[Control.OnInvalidate\(EventArgs\)](#) , [Control.OnKeyDown\(KeyEventEventArgs\)](#) ,
[Control.OnKeyPress\(KeyEventEventArgs\)](#) , [Control.OnKeyUp\(KeyEventEventArgs\)](#) ,
[Control.OnLeave\(EventArgs\)](#) , [Control.InvokeLostFocus\(Control, EventArgs\)](#) ,
[Control.OnLostFocus\(EventArgs\)](#) , [Control.OnMarginChanged\(EventArgs\)](#) ,
[Control.OnMouseDoubleClick\(MouseEventArgs\)](#) , [Control.OnMouseClick\(MouseEventArgs\)](#) ,
[Control.OnMouseCaptureChanged\(EventArgs\)](#) , [Control.OnMouseDown\(MouseEventArgs\)](#) ,
[Control.OnMouseEnter\(EventArgs\)](#) , [Control.OnMouseLeave\(EventArgs\)](#) ,
[Control.OnDpiChangedBeforeParent\(EventArgs\)](#) , [Control.OnDpiChangedAfterParent\(EventArgs\)](#) ,
[Control.OnMouseHover\(EventArgs\)](#) , [Control.OnMouseMove\(MouseEventArgs\)](#) ,
[Control.OnMouseUp\(MouseEventArgs\)](#) ,
[Control.OnQueryContinueDrag\(QueryContinueDragEventArgs\)](#) ,
[Control.OnRegionChanged\(EventArgs\)](#) , [Control.OnPreviewKeyDown\(PreviewKeyDownEventArgs\)](#) ,
[Control.OnSizeChanged\(EventArgs\)](#) , [Control.OnChangeUICues\(UICuesEventArgs\)](#) ,
[Control.OnSystemColorsChanged\(EventArgs\)](#) , [Control.OnValidating\(CancelEventArgs\)](#) ,
[Control.OnValidated\(EventArgs\)](#) , [Control.PerformLayout\(\)](#) , [Control.PerformLayout\(Control, string\)](#) ,
[Control.PointToClient\(Point\)](#) , [Control.PointToScreen\(Point\)](#) ,
[Control.PreProcessMessage\(ref Message\)](#) , [Control.PreProcessControlMessage\(ref Message\)](#) ,
[Control.ProcessKeyEventArgs\(ref Message\)](#) , [Control.ProcessKeyMessage\(ref Message\)](#) ,
[Control.RaiseDragEvent\(object, DragEventArgs\)](#) , [Control.RaisePaintEvent\(object, PaintEventArgs\)](#) ,
[Control.RecreateHandle\(\)](#) , [Control.RectangleToClient\(Rectangle\)](#) ,
[Control.RectangleToScreen\(Rectangle\)](#) , [Control.ReflectMessage\(nint, ref Message\)](#) ,
[Control.Refresh\(\)](#) , [Control.ResetMouseEventArgs\(\)](#) , [Control.ResetText\(\)](#) , [Control.ResumeLayout\(\)](#) ,
[Control.ResumeLayout\(bool\)](#) , [Control.Scale\(SizeF\)](#) , [Control.Select\(\)](#) ,
[Control.SelectNextControl\(Control, bool, bool, bool, bool\)](#) , [Control.SendToBack\(\)](#) ,
[Control.SetBounds\(int, int, int, int\)](#) , [Control.SetBounds\(int, int, int, int, BoundsSpecified\)](#) ,

[Control.SizeFromClientSize\(Size\)](#) , [Control.SetStyle\(ControlStyles, bool\)](#) , [Control.SetTopLevel\(bool\)](#) ,
[Control.RtlTranslateAlignment\(HorizontalAlignment\)](#) ,
[Control.RtlTranslateAlignment\(LeftRightAlignment\)](#) ,
[Control.RtlTranslateAlignment\(ContentAlignment\)](#) ,
[Control.RtlTranslateHorizontal\(HorizontalAlignment\)](#) ,
[Control.RtlTranslateLeftRight\(LeftRightAlignment\)](#) , [Control.RtlTranslateContent\(ContentAlignment\)](#) ,
[Control.Show\(\)](#) , [Control.SuspendLayout\(\)](#) , [Control.Update\(\)](#) , [Control.UpdateBounds\(\)](#) ,
[Control.UpdateBounds\(int, int, int, int\)](#) , [Control.UpdateBounds\(int, int, int, int, int, int\)](#) ,
[Control.UpdateZOrder\(\)](#) , [Control.UpdateStyles\(\)](#) , [Control.OnImeModeChanged\(EventArgs\)](#) ,
[Control.AccessibilityObject](#) , [Control.AccessibleDefaultActionDescription](#) ,
[Control.AccessibleDescription](#) , [Control.AccessibleName](#) , [Control.AccessibleRole](#) ,
[Control.AllowDrop](#) , [Control.Anchor](#) , [Control.AutoScrollOffset](#) , [Control.LayoutEngine](#) ,
[Control.DataContext](#) , [Control.BackgroundImage](#) , [Control.BackgroundImageLayout](#) ,
[Control.Bottom](#) , [Control.Bounds](#) , [Control.CanFocus](#) , [Control.CanRaiseEvents](#) ,
[Control.CanSelect](#) , [Control.Capture](#) , [Control.CausesValidation](#) ,
[Control.CheckForIllegalCrossThreadCalls](#) , [Control.ClientRectangle](#) , [Control.CompanyName](#) ,
[Control.ContainsFocus](#) , [Control.ContextMenuStrip](#) , [Control.Controls](#) , [Control.Created](#) ,
[Control.Cursor](#) , [Control.DataBindings](#) , [Control.DefaultBackColor](#) , [Control.DefaultCursor](#) ,
[Control.DefaultFont](#) , [Control.DefaultForeColor](#) , [Control.DefaultMargin](#) ,
[Control.DefaultMaximumSize](#) , [Control.DefaultMinimumSize](#) , [Control.DefaultPadding](#) ,
[Control.DeviceDpi](#) , [Control.IsDisposed](#) , [Control.Disposing](#) , [Control.Dock](#) ,
[Control.DoubleBuffered](#) , [Control.Enabled](#) , [Control.Focused](#) , [Control.Font](#) ,
[Control.FontHeight](#) , [Control.ForeColor](#) , [Control.Handle](#) , [Control.HasChildren](#) , [Control.Height](#) ,
[Control.IsHandleCreated](#) , [Control.InvokeRequired](#) , [Control.IsAccessible](#) ,
[Control.IsAncestorSiteInDesignMode](#) , [Control.IsMirrored](#) , [Control.Left](#) , [Control.Margin](#) ,
[Control.ModifierKeys](#) , [Control.MouseButtons](#) , [Control.mousePosition](#) , [Control.Name](#) ,
[Control.Parent](#) , [Control.ProductName](#) , [Control.ProductVersion](#) , [Control.RecreatingHandle](#) ,
[Control.Region](#) , [Control.RenderRightToLeft](#) , [Control.ResizeRedraw](#) , [Control.Right](#) ,
[Control.RightToLeft](#) , [Control.ScaleChildren](#) , [Control.Site](#) , [Control.TabIndex](#) , [Control.TabStop](#) ,
[Control.Tag](#) , [Control.Top](#) , [Control.TopLevelControl](#) , [Control.ShowKeyboardCues](#) ,
[Control.ShowFocusCues](#) , [Control.UseWaitCursor](#) , [Control.Visible](#) , [Control.Width](#) ,
[Control.PreferredSize](#) , [Control.Padding](#) , [Control.ImeMode](#) , [Control.ImeModeBase](#) ,
[Control.PropagatingImeMode](#) , [Control.BackColorChanged](#) , [Control.BackgroundImageChanged](#) ,
[Control.BackgroundImageLayoutChanged](#) , [Control.BindingContextChanged](#) ,
[Control.CausesValidationChanged](#) , [Control.ClientSizeChanged](#) ,
[Control.ContextMenuStripChanged](#) , [Control.CursorChanged](#) , [Control.DockChanged](#) ,
[Control.EnabledChanged](#) , [Control.FontChanged](#) , [Control.ForeColorChanged](#) ,
[Control.LocationChanged](#) , [Control.MarginChanged](#) , [Control.RegionChanged](#) ,
[Control.RightToLeftChanged](#) , [Control.SizeChanged](#) , [Control.TabIndexChanged](#) ,
[Control.TabStopChanged](#) , [Control.TextChanged](#) , [Control.VisibleChanged](#) , [Control.Click](#) ,

[Control.ControlAdded](#) , [Control.ControlRemoved](#) , [Control.DataContextChanged](#) ,
[Control.DragDrop](#) , [Control.DragEnter](#) , [Control.DragOver](#) , [Control.DragLeave](#) ,
[Control.GiveFeedback](#) , [Control.HandleCreated](#) , [Control.HandleDestroyed](#) ,
[Control.HelpRequested](#) , [Control.Invalidated](#) , [Control.PaddingChanged](#) , [Control.Paint](#) ,
[Control.QueryContinueDrag](#) , [Control.QueryAccessibilityHelp](#) , [Control.DoubleClick](#) ,
[Control.Enter](#) , [Control.GotFocus](#) , [Control.KeyDown](#) , [Control.KeyPress](#) , [Control.KeyUp](#) ,
[Control.Layout](#) , [Control.Leave](#) , [Control.LostFocus](#) , [Control.MouseClick](#) ,
[Control.MouseDoubleClick](#) , [Control.MouseCaptureChanged](#) , [Control.MouseDown](#) ,
[Control.MouseEnter](#) , [Control.MouseLeave](#) , [Control.DpiChangedBeforeParent](#) ,
[Control.DpiChangedAfterParent](#) , [Control.MouseHover](#) , [Control.MouseMove](#) , [Control.MouseUp](#) ,
[Control.MouseWheel](#) , [Control.Move](#) , [Control.PreviewKeyDown](#) , [Control.Resize](#) ,
[Control.ChangeUICues](#) , [Control.StyleChanged](#) , [Control.SystemColorsChanged](#) ,
[Control.Validating](#) , [Control.Validated](#) , [Control.ParentChanged](#) , [Control.ImeModeChanged](#) ,
[Component.Dispose\(\)](#) , [Component.GetService\(Type\)](#) , [Component.Container](#) ,
[Component.DesignMode](#) , [Component.Events](#) , [Component.Disposed](#) ,
[MarshalByRefObject.GetLifetimeService\(\)](#) , [MarshalByRefObject.InitializeLifetimeService\(\)](#) ,
[MarshalByRefObject.MemberwiseClone\(bool\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) ,
[object.ReferenceEquals\(object, object\)](#)

Constructors

Form1()

Sets up the canvas, parser, command facotry.

```
public Form1()
```

Methods

Dispose(bool)

Clean up any resources being used.

```
protected override void Dispose(bool disposing)
```

Parameters

disposing bool ↗

true if managed resources should be disposed; otherwise, false.

Class IfTest

Namespace: [ase assessment](#)

Assembly: unit_testing.dll

Represents testing for restricted and unrestricted If classes.

```
[TestClass]  
public class IfTest
```

Inheritance

[object](#) ← IfTest

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Methods

NoRestrictionTest()

Test the unrestricted If class. Ensures that multiple objects can be created.

```
[TestMethod]  
public void NoRestrictionTest()
```

Class IntTest

Namespace: [ase assessment](#)

Assembly: unit_testing.dll

Represents testing for restricted and unrestricted Int classes.

```
[TestClass]  
public class IntTest
```

Inheritance

[object](#) ← IntTest

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Methods

NoRestrictionTest()

Test the unrestricted Int class. Ensures that multiple objects can be created.

```
[TestMethod]  
public void NoRestrictionTest()
```

RestrictionTest()

Tests the restriction of the Int Class.

```
[TestMethod]  
public void RestrictionTest()
```

Class MethodTest

Namespace: [ase assessment](#)

Assembly: unit_testing.dll

Represents testing for restricted and unrestricted Int classes.

```
[TestClass]  
public class MethodTest
```

Inheritance

[object](#) ← MethodTest

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Methods

NoRestrictionTest()

Test the Custom method class.

```
[TestMethod]  
public void NoRestrictionTest()
```

Class RealTest

Namespace: [ase assessment](#)

Assembly: unit_testing.dll

Represents testing for restricted and unrestricted Real classes.

```
[TestClass]  
public class RealTest
```

Inheritance

[object](#) ← RealTest

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Methods

NoRestrictionTest()

Test the unrestricted Real class. Ensures that multiple objects can be created.

```
[TestMethod]  
public void NoRestrictionTest()
```

RestrictionTest()

Tests the restriction of the Real Class.

```
[TestMethod]  
public void RestrictionTest()
```

Class UnrestrictedArray

Namespace: [ase assessment](#)

Assembly: ase_assessment.dll

Represents the array class the inherits the base class called Array.

```
public class UnrestrictedArray : Array, ICommand
```

Inheritance

[object](#) ← Command ← Evaluation ← Array ← UnrestrictedArray

Implements

ICommand

Inherited Members

Array.PEEK , Array.POKE , Array.type , Array.rows , Array.columns , Array.valueInt , Array.valueReal ,
Array.intArray , Array.realArray , Array.pokeValue , Array.peekVar , Array.rowS , Array.columnS , Array.row ,
Array.column , Array.ArrayRestrictions() , Array.ReduceRestrictionCounter() , Array.Compile() ,
[Array.CheckParameters\(string\[\]\)](#) , Array.Execute() , [Array.ProcessArrayParametersCompile\(bool\)](#) ,
[Array.ProcessArrayParametersExecute\(bool\)](#) , [Array.SetIntArray\(int,int,int\)](#) ,
[Array.SetRealArray\(double,int,int\)](#) , [Array.GetIntArray\(int,int\)](#) , [Array.GetRealArray\(int,int\)](#) ,
Array.Rows , Array.Columns , Evaluation.expression , Evaluation.evaluatedExpression ,
Evaluation.varName , Evaluation.value , [Evaluation.ProcessExpression\(string\)](#) , Evaluation.Expression ,
Evaluation.VarName , Evaluation.Value , Evaluation.Local , Command.program , Command.parameterList ,
Command.parameters , Command.paramsint , [Command.Set\(StoredProgram, string\)](#) ,
[Command.ProcessParameters\(string\)](#) , Command.ToString() , Command.Program , Command.Name ,
Command.ParameterList , Command.Parameters , Command.Paramsint , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#)

Constructors

UnrestrictedArray()

Initializes the new instances of class.

```
public UnrestrictedArray()
```


Class UnrestrictedElse

Namespace: [ase assessment](#)

Assembly: ase_assessment.dll

Represents the unrestricted version of else class with reduced restrictions.

```
public class UnrestrictedElse : Else, ICommand
```

Inheritance

[object](#) ← Command ← Evaluation ← Boolean ← ConditionalCommand ← CompoundCommand ← Else ← UnrestrictedElse

Implements

ICommand

Inherited Members

[Else.CheckParameters\(string\[\]\)](#) , Else.Compile() , Else.Execute() , Else.CorrespondingEnd ,
CompoundCommand.ReduceRestrictions() , CompoundCommand.CorrespondingCommand ,
ConditionalCommand.endLineNumber , ConditionalCommand.EndLineNumber ,
ConditionalCommand.Condition , ConditionalCommand.LineNumber , ConditionalCommand.CondType ,
ConditionalCommand.ReturnLineNumber , Boolean.Restrictions() , Boolean.BoolValue ,
Evaluation.expression , Evaluation.evaluatedExpression , Evaluation.varName , Evaluation.value ,
[Evaluation.ProcessExpression\(string\)](#) , Evaluation.Expression , Evaluation.VarName , Evaluation.Value ,
Evaluation.Local , Command.program , Command.parameterList , Command.parameters ,
Command.paramsint , [Command.Set\(StoredProgram, string\)](#) , [Command.ProcessParameters\(string\)](#) ,
Command.ToString() , Command.Program , Command.Name , Command.ParameterList ,
Command.Parameters , Command.Parmsint , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) ,
[object.ReferenceEquals\(object, object\)](#)

Constructors

UnrestrictedElse()

Initializes the new instances of UnrestrictedElse.

```
public UnrestrictedElse()
```


Class UnrestrictedEnd

Namespace: [ase assessment](#)

Assembly: ase_assessment.dll

Represents the unrestricted version of class End with reduced restrictions.

```
public class UnrestrictedEnd : End, ICommand
```

Inheritance

[object](#) ← Command ← Evaluation ← Boolean ← ConditionalCommand ← CompoundCommand ← End ← UnrestrictedEnd

Implements

ICommand

Inherited Members

End.Compile() , End.Execute() , CompoundCommand.ReduceRestrictions() ,
[CompoundCommand.CheckParameters\(string\[\]\)](#) , CompoundCommand.CorrespondingCommand ,
ConditionalCommand.endLineNumber , ConditionalCommand.EndLineNumber ,
ConditionalCommand.Condition , ConditionalCommand.LineNumber , ConditionalCommand.CondType ,
ConditionalCommand.ReturnLineNumber , Boolean.Restrictions() , Boolean.BoolValue ,
Evaluation.expression , Evaluation.evaluatedExpression , Evaluation.varName , Evaluation.value ,
[Evaluation.ProcessExpression\(string\)](#) , Evaluation.Expression , Evaluation.VarName , Evaluation.Value ,
Evaluation.Local , Command.program , Command.parameterList , Command.parameters ,
Command.paramsint , [Command.Set\(StoredProgram, string\)](#) , [Command.ProcessParameters\(string\)](#) ,
Command.ToString() , Command.Program , Command.Name , Command.ParameterList ,
Command.Parameters , Command.Parmsint , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) ,
[object.ReferenceEquals\(object, object\)](#)

Constructors

UnrestrictedEnd()

Initializes the new instances of the Unrestricted class.

```
public UnrestrictedEnd()
```


Class UnrestrictedFor

Namespace: [ase assessment](#)

Assembly: ase_assessment.dll

Represents the unrestricted version of the For class. Allows the custom behaviour overriding Restrictions method.

```
public class UnrestrictedFor : For, ICommand
```

Inheritance

[object](#) ← Command ← Evaluation ← Boolean ← ConditionalCommand ← For ← UnrestrictedFor

Implements

ICommand

Inherited Members

For.Compile() , For.Execute() , For.LoopControlV , For.From , For.To , For.Step ,
ConditionalCommand.endLineNumber , ConditionalCommand.EndLineNumber ,
ConditionalCommand.Condition , ConditionalCommand.LineNumber , ConditionalCommand.CondType ,
ConditionalCommand.ReturnLineNumber , Boolean.BoolValue , Evaluation.expression ,
Evaluation.evaluatedExpression , Evaluation.varName , Evaluation.value ,
[Evaluation.CheckParameters\(string\[\]\)](#) , [Evaluation.ProcessExpression\(string\)](#) , Evaluation.Expression ,
Evaluation.VarName , Evaluation.Value , Evaluation.Local , Command.program , Command.parameterList ,
Command.parameters , Command.paramsint , [Command.Set\(StoredProgram, string\)](#) ,
[Command.ProcessParameters\(string\)](#) , Command.ToString() , Command.Program , Command.Name ,
Command.ParameterList , Command.Parameters , Command.Paramsint , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#)

Constructors

UnrestrictedFor()

Initializes the new instances of class Unrestricted.

```
public UnrestrictedFor()
```

Methods

Restrictions()

Implements the custom behaviour by overriding Restrictions method.

```
public override void Restrictions()
```

Class UnrestrictedIf

Namespace: [ase assessment](#)

Assembly: ase_assessment.dll

Represents the unrestricted version of the If class with reduce restrictions.

```
public class UnrestrictedIf : If, ICommand
```

Inheritance

[object](#) ← Command ← Evaluation ← Boolean ← ConditionalCommand ← CompoundCommand ← If ← UnrestrictedIf

Implements

ICommand

Inherited Members

CompoundCommand.ReduceRestrictions() , [CompoundCommand.CheckParameters\(string\[\]\)](#) ,
CompoundCommand.Compile() , CompoundCommand.CorrespondingCommand ,
ConditionalCommand.endLineNumber , ConditionalCommand.Execute() ,
ConditionalCommand.EndLineNumber , ConditionalCommand.Condition ,
ConditionalCommand.LineNumber , ConditionalCommand.CondType ,
ConditionalCommand.ReturnLineNumber , Boolean.BoolValue , Evaluation.expression ,
Evaluation.evaluatedExpression , Evaluation.varName , Evaluation.value ,
[Evaluation.ProcessExpression\(string\)](#) , Evaluation.Expression , Evaluation.VarName , Evaluation.Value ,
Evaluation.Local , Command.program , Command.parameterList , Command.parameters ,
Command.paramsint , [Command.Set\(StoredProgram, string\)](#) , [Command.ProcessParameters\(string\)](#) ,
Command.ToString() , Command.Program , Command.Name , Command.ParameterList ,
Command.Parameters , Command.Paramsint , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) ,
[object.ReferenceEquals\(object, object\)](#)

Constructors

UnrestrictedIf()

Initailzes the new instances of the UnrestrictedIf class.

```
public UnrestrictedIf()
```

Methods

Restrictions()

Implements the custom behaviour by overriding Restrictions method.

```
public override void Restrictions()
```

Class UnrestrictedInt

Namespace: [ase assessment](#)

Assembly: ase_assessment.dll

Represents the child class which inherits parent class Int.

```
public class UnrestrictedInt : Int, ICommand
```

Inheritance

[object](#) ← Command ← Evaluation ← Int ← UnrestrictedInt

Implements

ICommand

Inherited Members

Int.Compile() , Evaluation.expression , Evaluation.evaluatedExpression , Evaluation.varName ,
Evaluation.value , [Evaluation.CheckParameters\(string\[\]\)](#) , [Evaluation.ProcessExpression\(string\)](#) ,
Evaluation.Expression , Evaluation.VarName , Evaluation.Value , Evaluation.Local , Command.program ,
Command.parameterList , Command.parameters , Command.paramsint ,
[Command.Set\(StoredProgram, string\)](#) , [Command.ProcessParameters\(string\)](#) , Command.ToString() ,
Command.Program , Command.Name , Command.ParameterList , Command.Parameters ,
Command.Paramsint , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#)

Methods

Execute()

Executes the base class functionality.

```
public override void Execute()
```

Restrictions()

Defines the restriction for a class This method is intended to overriden in child class.

```
public override void Restrictions()
```

Class UnrestrictedReal

Namespace: [ase assessment](#)

Assembly: ase_assessment.dll

Represents the real class that inherit parent class. Provides the implementation of Execution and Restrictions.

```
public class UnrestrictedReal : Real, ICommand
```

Inheritance

[object](#) ← Command ← Evaluation ← Real ← UnrestrictedReal

Implements

ICommand

Inherited Members

Real.Compile() , Real.Value , Evaluation.expression , Evaluation.evaluatedExpression , Evaluation.varName , Evaluation.value , [Evaluation.CheckParameters\(string\[\]\)](#) , [Evaluation.ProcessExpression\(string\)](#) , Evaluation.Expression , Evaluation.VarName , Evaluation.Local , Command.program , Command.parameterList , Command.parameters , Command.paramsint , [Command.Set\(StoredProgram, string\)](#) , [Command.ProcessParameters\(string\)](#) , Command.ToString() , Command.Program , Command.Name , Command.ParameterList , Command.Parameters , Command.Paramsint , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#)

Methods

Execute()

Calls the primarily functionality of the real component.

```
public override void Execute()
```

Restrictions()

Defines the restrictions for the real class.

```
public override void Restrictions()
```

Class UnrestrictedWhile

Namespace: [ase assessment](#)

Assembly: ase_assessment.dll

Represents the Unrestricted version of the While class. Additionally, it contains reduced restrictions for more flexibility.

```
public class UnrestrictedWhile : While, ICommand
```

Inheritance

[object](#) ← Command ← Evaluation ← Boolean ← ConditionalCommand ← CompoundCommand ← While ← UnrestrictedWhile

Implements

ICommand

Inherited Members

CompoundCommand.ReduceRestrictions() , [CompoundCommand.CheckParameters\(string\[\]\)](#) ,
CompoundCommand.Compile() , CompoundCommand.CorrectingCommand ,
ConditionalCommand.endLineNumber , ConditionalCommand.Execute() ,
ConditionalCommand.EndLineNumber , ConditionalCommand.Condition ,
ConditionalCommand.LineNumber , ConditionalCommand.CondType ,
ConditionalCommand.ReturnLineNumber , Boolean.BoolValue , Evaluation.expression ,
Evaluation.evaluatedExpression , Evaluation.varName , Evaluation.value ,
[Evaluation.ProcessExpression\(string\)](#) , Evaluation.Expression , Evaluation.VarName , Evaluation.Value ,
Evaluation.Local , Command.program , Command.parameterList , Command.parameters ,
Command.paramsint , [Command.Set\(StoredProgram, string\)](#) , [Command.ProcessParameters\(string\)](#) ,
Command.ToString() , Command.Program , Command.Name , Command.ParameterList ,
Command.Parameters , Command.Paramsint , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) ,
[object.ReferenceEquals\(object, object\)](#)

Constructors

UnrestrictedWhile()

Initializes the new instances of UnrestrictedWhile class.

```
public UnrestrictedWhile()
```

Methods

Restrictions()

Defines the restriction for a class. This method is intended to be overridden in derived classes.

```
public override void Restrictions()
```

Class WhileTest

Namespace: [ase assessment](#)

Assembly: unit_testing.dll

```
[TestClass]
public class WhileTest
```

Inheritance

[object](#) ← WhileTest

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Methods

NoRestrictionTest()

Test the unrestricted While class. Ensures that multiple objects can be created.

```
[TestMethod]
public void NoRestrictionTest()
```

Namespace unit_testing

Classes

[AppCanvasTest](#)

Represents the test for AppCanvas Class.

[AppTriangleTest](#)

Represents the testing for the Triangle class.

Class AppCanvasTest

Namespace: [unit testing](#)

Assembly: unit_testing.dll

Represents the test for AppCanvas Class.

```
[TestClass]  
public class AppCanvasTest
```

Inheritance

[object](#) ← AppCanvasTest

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Methods

BoundaryMoveToTest()

Tests the method with boundary 0,0

```
[TestMethod]  
public void BoundaryMoveToTest()
```

FailingCircleTest()

Tests the Circle method with Invalid coordinates. Canvas Exception is expected.

```
[TestMethod]  
[ExpectedException(typeof(CanvasException))]  
public void FailingCircleTest()
```

FailingDrawToTest()

Tests the DrawTo method with Invalid coordinates. Canvas Exception is expected.

```
[TestMethod]
[ExpectedException(typeof(CanvasException))]
public void FailingDrawToTest()
```

FailingMoveToTest()

Tests the MoveTo method with Invalid coordinates. Canvas Exception is expected.

```
[TestMethod]
[ExpectedException(typeof(CanvasException))]
public void FailingMoveToTest()
```

FailingMultilineTesting()

Tests the Multiline with Invalid coordinates. Canvas Exception is expected.

```
[TestMethod]
[ExpectedException(typeof(CanvasException))]
public void FailingMultilineTesting()
```

FailingRectangleTest()

Tests the Rect method with Invalid coordinates. Canvas Exception is expected.

```
[TestMethod]
[ExpectedException(typeof(CanvasException))]
public void FailingRectangleTest()
```

PassingCircleTest()

Tests the Circle method with valid coordinates.

```
[TestMethod]
public void PassingCircleTest()
```

PassingDrawToTest()

Tests the DrawTo method with valid coordinates.

```
[TestMethod]
public void PassingDrawToTest()
```

PassingMoveToTest()

Tests the MoveTo method with valid coordinates.

```
[TestMethod]
public void PassingMoveToTest()
```

PassingMultilineTesting()

Tests the Multiline with valid coordinates.

```
[TestMethod]
public void PassingMultilineTesting()
```

PassingRectangleTest()

Tests the Rect method with valid coordinates.

```
[TestMethod]
public void PassingRectangleTest()
```

Class AppTriangleTest

Namespace: [unit testing](#)

Assembly: unit_testing.dll

Represents the testing for the Triangle class.

```
[TestClass]
public class AppTriangleTest
```

Inheritance

[object](#) ← AppTriangleTest

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Methods

FailingTriangleTest()

Tests the Tri method with Invalid coordinates. Canvas Exception is expected.

```
[TestMethod]
[ExpectedException(typeof(CanvasException))]
public void FailingTriangleTest()
```

PassingTriangleTest()

Test the Tri method with valid parameters.

```
[TestMethod]
public void PassingTriangleTest()
```