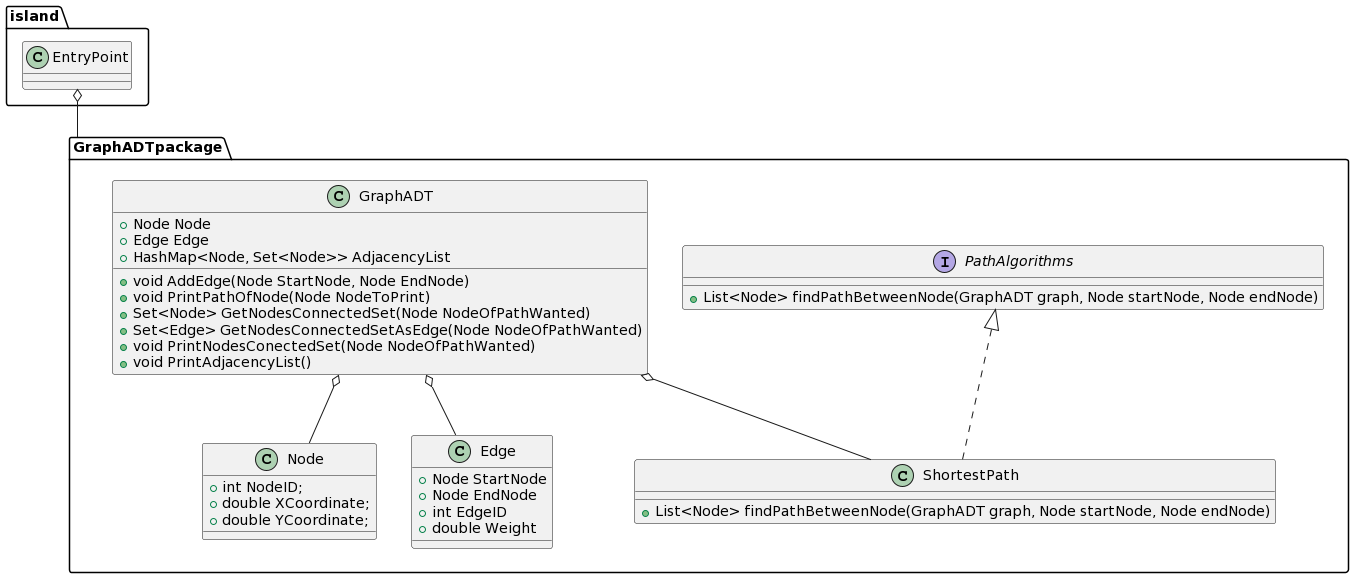
# Questions

1. Provide a class diagram of your pathfinder library



1. Explain why your code is SOLID and locate your technical debt. If you have used any pattern, describe them and justify your decision.
   1. For my code, I followed SOILD by first using single responsibility. Each class in my Graph Package only has one responsibility. Additionally,
2. Explain why your tests are “reasonable” (refer to CORRECT/BICEP if needed

First, I used the Ordering part of CORRECT to make sure my testing is reasonable. I ordered it by doing nodes, edges, Graph, Path. This means I work my way up the

1. Which design patterns have you used to support integrating your pathfinding library into your generator?

For my path finder class, I used the abstract factory pattern. We have an interface this means that we can create shortest path or any path algorithm that we want in the future without caring about how it is done. In the future, we will add methods and algorithms that we want into the interface what the interface will use. We have multiple ways of find the path, but we do not care how or which one to choose from.

For my GraphADT class, I to implement the singleton pattern. This means that we have one Graph for our whole Mesh. The GraphADT never will create a second graph that it can use unless it needs to do unit tests. I tried to implement it by only ever creating one instance of the graph in my mesh. This make it so that we only create the Graph when needed and our graph and adjacency list is encapsulated into our graph.

1. What about performances? Do you envision any scalability issues in your road network generation?

For performance, I went with an adjacency list to represent my graph. I first went with a matrix representation, but as soon as I added my nodes; it created a huge matrix that took up way to much memory. This meant I decided to go with an adjacency list which would provide much better performance on my graph. Likewise, this means that as we scale up the product in terms of graph size, we can take on any size graph and path needed. However, if we were to add the bonus or extra features, we would need a quite a bit of refactoring. But overall, this scalability really depends on what we are adding. I believe that reasonably sized scalable tasks would be easy to do, and I am prepared for that in my work.

I honestly do not see any scabaility problems with my road network. I have a Graph system that is able to handle any sized graph. Tha

1. Draw a high-level sequence diagram explaining how your generation process works from a coarse-grained point of view. The point is to document the whole process, not only the urbanism part.
2. Locate your technical debt in the generator. How can one extend it by adding a new generation mechanism in the generator? For example, if one plans to introduce resource production mechanisms (e.g., forests produce wood, lakes produce fish, …), how would it be supported by your project?

# Self-Reflection

1. Backward: What process did you go through to produce this result?

To produce the result that I wanted I had to fail a lot. Failing meant that I wrote a lot of iterations of the same code, tested it, and tried again as it fundamentally didn’t work. I started with my Graph ADT being completely wrong and way to convoluted. So, from the code I wrote I scrapped everything I did and started again. The main process I learned from was “fail early, fail often”. The process also included planning. There were a lot of things I didn’t expect and through this failed a lot.

1. Inward: What were your standards for this piece of work? Did you meet your standards?

I think what I produced met my standards. My main standard for the work is that it worked to create a mesh with the embellishments and our code had good OOP structure. Our first section was not really that good, but my Assignment 3 it really improved. I can always go back and redo a lot of the code that seems obvious to fix now that there is no grade attached. This means although some parts don’t meet my standard now, I am able to go back and fix it.

1. Outward: What is the one thing you particularly want people to notice when they look at your work?

I think the different features we add will be the main thing. In A3, we were able to get the bonus done. This means that there are multiple ways someone can view our graph. This aspect is cool because it takes the same mesh and creates something that is completely different. Adding all the arguments will also be something that stands out. Customizing the mesh will be something that I want the user to notice.

1. Forward: What lessons will you keep from this assignment in your professional practice?

From this assignment I learned a lot. I really understand how OOP works and what it means to have good practice when creating code. Also, I learned a lot about design software and how we can implement these ideas into our own work. Likewise, I really learned the importance of data structure and how you should always think before deciding what structure you want to use.