Solution to Homework
Advantages and disadvantages of ADT

- Advantages:
  1. Preserve/control the integrity of a class’ data
     - We do this by defining a class invariant
       - Definition: A class invariant is a statement (part of class documentation) describing characteristics of a class that do not vary for all objects instantiated from this class
       - For example:
         All temperature objects instantiated from the Temperature class always have their attribute myDegrees >= ABSOLUTE_ZERO_CELSIUS if their myScale == 'C' or have their attribute myDegrees >= ABSOLUTE_ZERO_FAHRENHEIT if their myScale == 'F'
Solution to Homework

Advantages and disadvantages of ADT

- Advantages:
  2. Ease s/w development team work
    - Software developer A writes the data collection class as an ADT:
      1. **visible** (public) section
         - Public Interface
      2. **hidden** (private) section:
         - Underlying data structure
         - Implementation of the ADT’s operations
    - Software developer B writes the client code, i.e., classes using this ADT class, simply by using the ADT’s public interface
    - As soon as Software developer A has created the ADT’s public interface, Software developer B can start development – all working at the same time on their own section of the s/w project (in parallel)
Solution to Homework

Advantages and disadvantages of ADT

Advantages:

3. Ease modification
   - Software developer A can change the ADT’s hidden (private) section
     - Underlying data structure
     - Implementation of the ADT’s operations
   - Without affecting the ADT’s public interface and therefore without affecting Software developer B’s client code
   - We can modify the implementation of a class (what is defined as private – behind the wall) without affecting client code, i.e., making changes to the private section of a class does not (should not) require us to make changes to the public section of a class
Solution to Homework

Advantages and disadvantages of ADT

Advantages:

4. Reduce complexity (from a client code’s perspective)

- In order to instantiate and use objects of an class (implemented as an ADT), all we (writing client code) need to know is the class’ public interface (how to call the class’ public methods), we do not need to the class’ implementation.
Solution to Homework
Advantages and disadvantages of ADT

 Advantages:
  5. Ease reusability
     Can be application-independent, hence reusable in another application
     Data collection List ADT can be reused in another application whenever a List is needed without having to modify, recompile and retest the code

Note:
 Right now, we cannot fully achieve this advantage because we need to modify the code, i.e., changing the data type of the data we store in our data collection, recompile and retest our List ADT class for every application
 However, we shall soon see “template”, which is a mechanism that will allow us to completely achieve the advantage described above
Solution to Homework

Advantages and disadvantages of ADT

- Disadvantages:
  1. More code: A class, implemented as an ADT, must offer getters/setters methods and methods that allow client code to access the class’ attributes
  - More code means:
    - More testing
    - More possibility for bugs to be introduced into our code
Solution to Homework

Advantages and disadvantages of ADT

- Disadvantages:

  2. Limited access: We, as user of the class (as writer of client code that makes use of the class), may feel limited by the class’ public methods

    - There may be operations we would like to do “on” objects of this class, operations that are not offered in the class’ public interface