Text

Description automatically generated Text

Description automatically generated Text

Description automatically generated



Table

Description automatically generated



Text

Description automatically generated Text

Description automatically generated Text

Description automatically generated Text

Description automatically generated

Text

Description automatically generated



* Recursion breaks a problem into smaller identical problems

Text, letter

Description automatically generated

* Each recursive call solves an identical but smaller problem
* A static method is a class method
* Non-text files are called binary or general files
* Text file is a file of characters organized into lines
* Invariant -> A condition that is always true at a particular point
* Loop Invariant -> A condition which is true before and after the execution of each loop
* A solution is good if the total cost incurred over all phases of it’s life cycle is minimal
* Encapsulation -> Objects combine data and operations

 

* Inheritance: Classes can inherit properties, generally behavior & structure from other classes
* Polymorphism -> Objects can determine appropriate operations at execution time
* Abstract Data Type(ADT) -> A collection of data and a set of operations on the data
* An ADT’s operations can be used without knowing how the operations are implemented. However, operation specifications should be known.

Text, letter

Description automatically generated

* Within problem solving, ADT’s support algorithms and algorithms are part of what constitute ADT’s
* All modules and ADT’s should hide something
* Encapsulation hides inner details
* Unified Modeling Language (UML) is a modeling language used to express object-oriented designs
* Modularity isolates errors
* Constructor -> Is a special kind of a method which has the same name as the class and no return type. Is executed only when an object is crated.
* Constructors allocate memory for an object and can initialize an object’s data

A close relationship exists between mathematical induction and recursion as induction can be used to prove the properties of a recursive algorithm

A grammar is a device for defining a language. Grammars are frequently recursive

A language is a set of strings of symbols. A recognition algorithm for a language can often be based directly on the grammar of a language

A class’s data fields may or may not be public. However, making them private is a recommended best practice. Also, by default all members in a class are private unless a programmer specifies otherwise.

* Default constructor has no parameters and typically initializes data fields to values the class implementation chooses
* Compiler generated default constructor: generated by the compiler if no constructor is included in a class
* For and While loops are equivalent. However, For is much more used and preferred than the While loop.
* Files provide both sequential and random access; Arrays only provide random access
* Files grow; Arrays remain constant with regards to size.
* Selection Statements -> if, switch; Iteration Statements -> while, for, and do-while
* String Tokenizer -> Strings that can be broken into pieces called tokens

Modularity keeps the complexity of a large program manageable by systematically controlling the interaction of its components. Also, isolates errors and eliminates redundancies

Procedural abstraction separates the purpose and use of a module from its implementation

A module’s specifications should detail how the module behaves and identify details that can be hidden within the module

Client: A program or a module that uses the class

Coding without a solution design increases debugging time

* Modules -> Self-contained units of code. Are designed to be loosely connected and highly cohesive.

A sub-class inherits the members of its super class

Inheritance enables the reuse of existing classes

Inheritance reduces unnecessary efforts to add additional features to an existing object

A sub-class can add new members to those it inherits

A sub-class can override inherited method of its superclass

* Methods should check their invariants and enforce their pre-conditions.
* Object-Oriented Programming (OOP) allows re-use of existing classes, thus saving development time.
* Specifications do not indicate how to implement a method.
* A modular program is easier to write, read, and modify
* An ADT is a collection of data and a set of operations on that data
* Data structures are part of an ADT’s implementation
* ADT and data structures are not same
* The ADT sorted list maintains item in sorted order. It also inserts and deletes items by their values, not their positions
* You can use ADT operations in an application without the distraction of implementation details

An instance of sub-class has all the behaviors of its superclass

A sub-class inherits private members from the super class, but cannot access them directly

A sub-class method can call a public superclass method

Clients of the sub-class can invoke the public methods of the superclass

Object Oriented Design (OOD) -> Produces modular solutions for problems that primarily involve data. Identifies objects by focusing on the nouns in the problem statement

Functional Decomposition (FD) -> Produces modular solutions for the problems in which the emphasis is on the algorithms. Identifies actions by focusing on the verbs in the problem statement. In FD, a task is addressed successively at lower levels of detail.

Use OOD and FD in tandem to formulate modular solutions.

Procedural Abstraction -> Specifies what to do, not how to do it. Separates the purpose of a method from its implementation. Each method will be/should be specified before it is implemented.

Data Abstraction -> Specify what you will do to data, and not how to do it.

Implicit Conversions/Integral Promotions -> If the operands of an operator differ in data types, the data type that is lower in the hierarchy is converted to one that is higher. Example: int -> long -> float -> double

* Encapsulation hides implementation details
* Java package provides a way to group related classes together
* Access to a package’s classes can be public or restricted
* “super” keyword -> Used in a constructor of a subclass to call the constructor of the superclass
* ADT list contains some operations such as insert, delete, and retrieve
* A reference variable as a data field of a class has the default value null
* A local reference variable has no default value
* The “new” is a Java keyword
* An array of objects is actually an array of references to the objects
* A traverse operation visits each node in the linked list
* Equality operators compare values of reference variables, not the objects that they reference. The “equals” method compares objects field by field
* A “.java” file cannot have more than one public class
* A new node can be inserted into a linked list
* A specified node can be deleted from a linked list

Reuse of an already implemented class saves time

Parentheses, operator precedence, and left-to-right association determine where to place operators in the post-fix expression

Stack can be used to organize an exhaustive search

Backtracking: A strategy for guessing at a solution and backing up when an impasse is reached

Recursion and backtracking can be combined for problem solving

A Java identifier begins with a letter and is followed by zero or more letters and digits

The ADT stack can be implemented using an array, a linked list, or an ADT list

StackInterface provides a common specification for the three implementations

A client can reference a class interface instead of the class itself

An interface specifies behavior that are common to a group of classes

Final or static methods use static binding

A generic class describes a class in terms of data-type parameter

A client specifies an actual data type when declaring an instance of a class

Primitive types are not allowed as type parameters

Instances of generic classes are not related

Text

Description automatically generated A picture containing text, indoor

Description automatically generated Text

Description automatically generatedText

Description automatically generated

Private data fields are hidden from the client

StackException provides simple ways to indicate unusual events

Instances of an array-based ADT implementation cannot contain items of a primitive type as they are not instances of the Object class. A corresponding wrapper class is required for their usage

Text

Description automatically generatedText

Description automatically generatedTable

Description automatically generatedText

Description automatically generatedText

Description automatically generated

Diagram

Description automatically generated

Table

Description automatically generated Text, letter

Description automatically generated Text, letter

Description automatically generated Text

Description automatically generated Text, letter

Description automatically generated Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated Text

Description automatically generated

A class with no access modifier is available to other classes in the same package

Inheritance should imply an “is-a” relationship

Another name for “has-a” relationship is containment

You can use instances of a sub-class anywhere you can use the instances of the super class

Text

Description automatically generated with medium confidence

Text, letter

Description automatically generated Text

Description automatically generated

A class that contains at least one abstract method is an abstract class

An abstract class has sub-classes but no instances

The extends clause places an upper bound on the data type parameter

Super clause places a lower bound on the data type parameter

An iterator accesses a collection one item at a time

A page of a book

Description automatically generated with low confidence

Text

Description automatically generated

Graphical user interface, text, email

Description automatically generated Text

Description automatically generated

A page of a book

Description automatically generated with low confidence Text, letter

Description automatically generated

Text, letter

Description automatically generated

Graphical user interface, text

Description automatically generated

