**Family Tree System Report**

**Corey Black**

The FamilyTree program is designed to simulate a family tree data structure where family members can be added and displayed. The program allows the user to set up an initial ancestor and partner, add children, add partners to existing family members, and display the family tree structure.

The program consists of two main classes: **FamilyTree** and **FamilyTreeNode**. The **FamilyTree** class represents the entire family tree and provides methods for adding family members and displaying the family structure. The **FamilyTreeNode** class represents individual family members and their relationships.

**Features:**

1. **Creation of an initial ancestor and partner.**

The program allows the user to initialize the family tree by defining an initial ancestor and their partner. This establishes the foundation upon which subsequent family members will be added. The FamilyTree class provides the means to set up these relationships, ensuring a structured and organized family tree.

1. **Addition of children to existing family members.**

The user can expand the family tree by adding children to existing family members. This feature enables the representation of generational progression within the family structure. By specifying parent-child relationships, the user can accurately observe the lineage of the family tree.

1. **Addition of partners to existing family members.**

Family dynamics often involve multiple partnerships and marriages. The program accommodates this complexity by allowing users to add partners to existing family members. This ensures the user can accurately reflect on the diverse nature of familial connections. The Family Tree class provides methods to establish partner relationships.

1. **Display of the family tree structure.**

A vital part of the program is to visually represent the hierarchical structure of the family tree. The family tree class offers methods to generate clear and informative visualizations.

**Design and Implementation**

1. FamilyTree Class

The family tree class is responsible for managing the overall family tree structure, it encapsulates methods for adding family members, establishing relationships, and generating visual representations. This class is the central hub for interacting with the family tree data structure.

1. FamilyTreeNode Class

Each member of the family tree is represented as a FamilyTreeNode object. These objects store information about the individual, e.g.; their name, gender, and relationships with other members of the family. Organizing the objects into nodes ensures easy identification and traversal of the family tree structure.

**Testing:**

**Scenario 1: Creating a Family Tree**

1. Create a **FamilyTree** object with an ancestor name "James" and a partner name "Mary".
2. Verify that the program prints a message indicating the successful creation of the family tree.

**Scenario 2: Adding Children**

1. Add children "John", "Amy", and "Alan" to the ancestor "James".
2. Verify that the program correctly adds these children to the family tree structure.

**Scenario 3: Adding Partners**

1. Add partners "Alice" to "John" and "Emily" to "Alan".
2. Verify that the program correctly adds these partners to the respective family members.

**Scenario 4: Displaying Family Structure**

1. Display the entire family tree structure.
2. Verify that the program correctly displays the ancestor, partners, and children in the family tree.

**Family Tree System Testing**

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Expected Output | Actual Output | Pass/Fail |
| Creating a Family Tree | Family tree is created successfully | Family tree created successfully. | Pass |
| Add Children | Children are added successfully | Children was added successfully. | Pass |
| Adding Partners | Partners are added successfully | Partners are added successfully. | Pass |
| Displaying Family Structure | - James (identifier 1) partner Mary (identifier 2)<br> Children:<br> - John (identifier 3)<br> - Amy (identifier 4)<br> - Alan (identifier 5)<br> <br> | - James (identifier 1) partner Mary (identifier 2)<br> Children:<br> - John (identifier 3)<br> - Amy (identifier 4)<br> - Alan (identifier 5)<br> <br> | Pass |
| Displaying Specific Family Member (amy) | Displaying specific family member:  Amy (identifier 4) partner none  Children: | Displaying specific family member:  Amy (identifier 4) partner none  Children: | Pass |