Program name : Project2_Design_Description_Schaefer_Kristin.pdf
Author : Kristin Schaefer
Date : 04-29-2019

Design for Lab3

Main Function

Description: The main function is used to the create the arrays of Animal objects to pass to the Zoo constructor, to call the Zoo menu functions, and to play the Zoo game while the user chooses to keep playing.

Process:

1. Create arrays of Tiger, Turtle and Penguin objects
2. Create Zoo object, calling constructor and passing Tiger, Turtle and Penguin arrays to pointers in Zoo
3. Call menuStart function
4. If user selects to start Zoo Tycoon game, call playZooTycoon function and start game
4. Call menuInput function and ask for input of number of animals to purchase
5. Based on user input, update number of animals, age of animals (1 year old), account balance, increment days.
6. Start loop which is based on player's account balance. If player's balance is negative at end of day, end game.
7. For each iteration of the loop (which represents one day of Zoo operations)
    i. increment each animal's age by 1 day
    ii. deduct feeding costs from player's account balance
    iii. select one random event to occur
    iv. calculate profit for the day based on number of animals and their payoff costs and add to player's account balance
    v. if there is a bonus add to player's account balance
    vi. ask the player if they would like to buy an adult animal
    vii. if the player would like to buy an animal, ask which type of animal, deduct cost, and update age to 3 days old.
    viii. increment the number of days
    ix. ask the user to if they would like to keep playing the game.
    x. if the user has no money, print a message to tell the user the game is over and end the game

Zoo Class

Description: The Zoo class contains all of the functions related to playing the Zoo Tycoon game. It has a menu function to start the game and a menu function to continue playing the game. It has several functions and variables related to playing the game, such as keeping track of the user's balance, the number of Animals owned and their associated daily costs, and implementing a daily random event such as the birth of an Animal.

Key data members:

1. int, numTigers, numTurtles, numPenguins
    - description
2. Tiger *tigerPtr, Turtle *turtlePtr, Penguin *penguinPtr
    - pointers for dynamic arrays of animal objects
    - set to nullptr in private member variables
2. int days
    - describes the number of days the Zoo is in operation
3. double balance
    - holds the player's account balance
    - initial balance is set to 100,000
4. int tigerArraySize, turtleArraySize, penguinArraySize
    - keeps track of the array size for using dynamic arrays

Key functions:

1. Zoo constructor
    - sets pointers to Animal object arrays passes as parameters
2. menuStart
    - 1. Play Zoo Tycoon
    - 2. Quit
3. menuInput
    - Get and set number of animals

4. playZooTycoon
    -
    - implements all of the daily zoo functions
5. randomEvent
    - generates a random event integer 1, 2, 3 or 4
    - implements one of the random event functions based on the random integer produced
6. animalBorn
    - selects a random Animal
    - determines if age is sufficient to have baby
    - creates new Animal objects if an Animal is born
6. Zoo deconstructor
    - deallocate memory of Tiger, Turtle and Penguin pointers

Animal Class

Description: The Animal class is the parent class of the derived classes Tiger, Turtle and Penguin. It contains protected variables such as the age of the Animal, it's cost, the number of babies it can have, it's food cost, and the amount of payoff. It has one virtual member function getPayoff.

Key data members:

1. int age
2. double cost
3. int numBabies
4. double baseFoodCost
5. double payoff

Key functions:

1. Animal constructor
    - sets the values for age, cost, numBabies, baseFoodCost, payoff
2. int getAge
    - returns an integer indicating the age of the Animal
3. setAge(int)
    - takes an int indicating the number of days to increase the Animal's age by
3. bool isAdult
    - returns a boolean value indicating if the Animal is an adult or baby
    - informs Zoo class if Animal is able to have baby or not
4. double getCost
    - returns a double indicating the cost of the Animal
    - used to update the balance variable in the Zoo class
5. int getNumBabies
    - returns an integer indicating the number of babies the Animal has
    - informs whether the array needs to be resized or not
6. int getBaseFoodCost
    - returns a double indicating the cost of food per day for the Animal
    - used to update the balance variable in the Zoo class
7. virtual double getPayoff();
    - returns a double indicating the payoff of the Animal per day
    - can be overidden by Tiger, Turtle and Penguin derived classes