```
//Savannah Swan
/*Class CS1450
* Due:Febuary 12, 2025
* Assignment 3
* This assignment reads from a file and places the values into a polymorphic array.
* Then it also works with an abstract class and interface to make subclasses include
* certain methods. The methods in main also involve working with the array to display and
* find the most talented animal and climbers. The subclasses implement the interfaces and extend
* the super class Animal.
*/
import java.io.File;
import java.util.ArrayList;
import java.io.IOException;
import java.util.Scanner;
public class SwanSavannahAssignment3 {
	public static void main(String[] args) throws IOException {

				final String FILE_NAME = "Animals.txt";

				File inputFileName = new File(FILE_NAME);

				Scanner inputFile = new Scanner (inputFileName);
				int numAnimals = inputFile.nextInt();

				Animal[] animal = new Animal[numAnimals];

				for (int i = 0; i < animal.length; i++) {

					String name = inputFile.next();
					String species = inputFile.next();
					int swimSpeed = inputFile.nextInt();
					int runSpeed = inputFile.nextInt();
					int climbSpeed = inputFile.nextInt();

					if (species.equals("alligator")) {
						Alligator myAlligator = new Alligator(name, species,
swimSpeed, runSpeed);
						animal[i] = myAlligator;
						}
					if (species.equals("bear")) {
						Bear myBear = new Bear(name, species, swimSpeed,
runSpeed, climbSpeed);
						animal[i] = myBear;
						}
					if (species.equals("monkey")) {
						Monkey myMonkey = new Monkey(name, species, runSpeed,
climbSpeed);
						animal[i] = myMonkey;
						}
					if (species.equals("giraffe")) {
						Giraffe myGiraffe = new Giraffe(name, species, runSpeed);
						animal[i] = myGiraffe;
						}
					if (species.equals("sloth")) {
```

```
                                                Sloth mySloth = new Sloth(name, species, swimSpeed,
climbSpeed);
                                                animal[i] = mySloth;
                                                }
                                }//for

                                displayAnimal(animal);

                                System.out.println("------------------------------------------------");
                                System.out.println("         Animals That Can Climb ");
                                System.out.println("------------------------------------------------");
                                System.out.println("Name          Species         Climb Speed");
                                System.out.println("------------------------------------------------");

                        for (int i = 0; i < findClimbers(animal).size(); i++) {
                        System.out.print("\n" + findClimbers(animal).get(i).getName() + "\t\t" +
                                        findClimbers(animal).get(i).getSpecies()  + "\t\t" +
                                        ((someInterface) findClimbers(animal).get(i)).climb());
                        }

                                System.out.println("\n-----------------------------------------");
                                System.out.println("       Most Skilled Animal ");
                                System.out.println("-----------------------------------------");

                                System.out.println(animal[findMostSkilled(animal)].getName() + " the " +
                                                animal[findMostSkilled(animal)].getSpecies() + " says " +
                                                animal[findMostSkilled(animal)].makeNoise() + " \nSwim
Speed: " +

                                                (((someInterface) animal[findMostSkilled(animal)]).swim()) +
                                                " \nRun Speed: " +
                                                (((someInterface) animal[findMostSkilled(animal)]).run()) +
                                                " \nClimb Speed: " +
                                                (((someInterface) animal[findMostSkilled(animal)]).climb()));

        }//main

        public static void displayAnimal(Animal[] animal) {

                System.out.println("------------------------------------------------");
                System.out.println("             All Animals in Array                ");
                System.out.println("------------------------------------------------");

                for (int i = 0; i < animal.length; i++) {
                        System.out.println("\n" + animal[i].getName() +" the " + animal[i].getSpecies() + " says "
                + animal[i].makeNoise() + "\nRun Speed: " + ((someInterface) animal[i]).run()
                +  "\nSwim Speed: " + ((someInterface) animal[i]).swim() +
                "\nClimb Speed: " + ((someInterface) animal[i]).climb());

                }

                }//displayAnimal

                public static ArrayList<Animal> findClimbers (Animal[] animals) {
```

```
				ArrayList<Animal> animalList = new ArrayList<>();

				for (int i = 0; i < animals.length; i++) {
					if (((someInterface) animals[i]).climb() > 0) {
						animalList.add(animals[i]);
					}
				}
				return animalList;
			}

			public static int findMostSkilled(Animal[] animal)
			{
				int biggest = ((someInterface) animal[0]).swim() +
						((someInterface) animal[0]).climb() +
						((someInterface) animal[0]).run();
				int index = 0;

				for (int i = 1; i < animal.length; i++)
				{

					int num = ((someInterface) animal[i]).swim() +
							((someInterface) animal[i]).climb() +
							((someInterface) animal[i]).run();

					if (num > biggest)
					{
						biggest = num;
						index = i;
					}
				}
				return index;
			}

	}//SwanSavannahAssignment3
	interface someInterface {

		public abstract int swim();
		public abstract int run();
		public abstract int climb();

		}//someInterface
		abstract class Animal {

			private String name;
			private String species;

			public void setName(String name) {
				this.name= name;
			}
			public void setSpecies(String species) {
				this.species = species;
			}

			public String getName() {
```

```java
			return name;
		}
		public String getSpecies() {
			return species;
		}

		public abstract String makeNoise();

	}//Animal

	class Alligator extends Animal implements someInterface {

		private int swimSpeed;
		private int runSpeed;

		public Alligator(String name, String species, int swimSpeed, int runSpeed) {
			this.swimSpeed = swimSpeed;
			this.runSpeed = runSpeed;
			setName(name);
			setSpecies(species);
			}

		@Override
		public int run() {
			return runSpeed;
		}
		@Override
		public int swim() {
			return swimSpeed;
		}
		@Override
		public int climb() {
			return 0;
		}
		@Override
		public String makeNoise() {
			return " Crunch! ";
		}

	}//Alligator

	class Bear extends Animal implements someInterface {
		private int swimSpeed;
		private int runSpeed;
		private int climbSpeed;

		public Bear(String name, String species, int swimSpeed, int runSpeed, int climbSpeed) {
			this.swimSpeed = swimSpeed;
			this.runSpeed = runSpeed;
			this.climbSpeed = climbSpeed;
			setName(name);
			setSpecies(species);

			}
		@Override
		public int run() {
			return runSpeed;
		}
		@Override
		public int swim() {
			return swimSpeed;
		}
		@Override
		public int climb() {
			return climbSpeed;
		}
		@Override
		public String makeNoise() {
			return "Growl!";
		}

	}//Bear

	class Giraffe extends Animal implements someInterface {
		//private int swimSpeed;
		private int runSpeed;
		//private int climbSpeed;

		public Giraffe(String name, String species, int runSpeed) {
			this.runSpeed = runSpeed;
			setName(name);
			setSpecies(species);
			}

		@Override
		public int run() {
			return runSpeed;
		}

		@Override
		public String makeNoise() {
			return " Bleat! ";
		}

		@Override
		public int swim() {
			// TODO Auto-generated method stub
			return 0;
		}
		@Override
		public int climb() {
			// TODO Auto-generated method stub
			return 0;
		}

	}//Giraffe
```

```
class Monkey extends Animal implements someInterface {
        //private int swimSpeed;
        private int runSpeed;
        private int climbSpeed;

        public Monkey(String name, String species, int runSpeed, int climbSpeed) {
                this.runSpeed = runSpeed;
                this.climbSpeed = climbSpeed;
                setName(name);
                setSpecies(species);
                }

        @Override
        public int run() {
                return runSpeed;
        }
        @Override
        public int swim() {
                return 0;
        }
        @Override
        public int climb() {
                return climbSpeed;
        }
        @Override
        public String makeNoise() {
                return " Screech! ";
        }

}//Monkey

class Sloth extends Animal implements someInterface {
        private int swimSpeed;
        private int runSpeed;
        private int climbSpeed;

        public Sloth(String name, String species, int swimSpeed, int climbSpeed) {
                this.swimSpeed = swimSpeed;
                this.climbSpeed = climbSpeed;
                setName(name);
                setSpecies(species);
                }

        @Override
        public int run() {
                return 0;
        }
        @Override
        public int swim() {
                return swimSpeed;
        }
        @Override
        public int climb() {
                return climbSpeed;
        }
        @Override
        public String makeNoise() {
                return " Squeak! ";
        }

}//Sloth
```