```java
public interface Speakable
        public String speak()

public interface Walkable
        void walk()
        void walkToward(Human H)


public class Human implements Comparable {
        private String name;
        public int age
        public static int speciesCode=1357

        public Human(String n, int a) {
                name = n
                age = a
                creation()
        }

        public void setName(String n){
                name = n
        }

        public String getName(0) {
                return(name)
        }

        public void creation() {
                System.out.println("Human created")
        }

        public void creation(int x) {                  //code   }

        public int compareTo(Object o) {       //code compares based on age    }

        public static int getSpeciesCode() {
                return( speciesCode )
        }
}

public class Baby extends Human implements Speakable {

        public Baby(String n){
                super(n, 0)
                creation()
        }

        public void eat() {             //code   }

        public String speak() {         //code   }

        public void creation() {
                System.out.println("Baby created")
        }
}
```

```java
public class Kid extends Baby implements Walkable {
        public double height;

        public Kid(String n, int a, double h) {
                super(n)
                age=a
                height = h
                creation()
        }

        public void creation() {
                System.out.println("Kid created")
        }

        public String speak() {           //code   }

        public void walk() {              //code   }

        public void walkToward(Human H) {              //code   }

        public String toString(){
                System.out.println("Name: " + getName() )
                System.out.println("Age: " + age )
                System.out.println("Height: " + height )
        }

        public static int add(int a, int b) {
                return(a + b)
        }
}


public abstract Teen extends Kid {
        public double attitudeLevel
        private String highSchool;

        public Teen(String n, int a, double h) {
                super(n, a, h)
                creation()
        }

        abstract void doHomework(String s)
        abstract void bePolite()

        public void setHighSchool(String s) {
                highSchool=s
        }

        public String getHighSchool() {
                return(highSchool)
        }
}
```

```java
public class GoodTeen extends Teen  {

        public GoodTeen(String n, int a, double h) {
                super(n, a, h)
        }

        public void doHomework(String s) {              //code   }

        public void bePolite() {          //code   }

        public String speak() {           //code   }
}



public class BadTeen extends Teen {

        public BadTeen(String n, int a, double h) {
                super(n, a, h)
        }

        public void doHomework(String s) {              //code   }

        public void bePolite() {          //code   }

        public String speak() {           //code   }

        public void causeTrouble() {      //code    }
}



public class RandomStuff  {
        //all these methods are coded to do something that makes sense
        //some members and methods not shown
        public Human getHuman()
        public String locateHuman(Human H)
        public String locateTeen(Teen T)
        public Kid getKid()
        public void walk(Walkable W)
        public Comparable findLeast(Comparable C)
        public Speakable[] whocanTalk(Baby[] list)
        public void giveTreat(Kid K)
        public Baby makeBaby(Teen T1, TeenT2)
}
```