SUPERKLASSEN DYR:

package KollarOblig2;  
//SUPERKLASSEN DYR  
  
public class Dyr implements Comparable<Dyr>{  
 //OPPRETTER VARIABLER FOR DET DYRENE HAR FELLES:  
 private String ID;  
 private String kjonn;  
 private Float lengde;  
 private Float vekt;  
 private String dato;  
 private String sted;  
  
 //OPPRETTER KONSTRUKTØRENE:  
 public Dyr(String ID, String kjonn, Float lengde, Float vekt, String dato, String sted) {  
 this.ID = ID;  
 this.kjonn = kjonn;  
 this.lengde = lengde;  
 this.vekt = vekt;  
 this.dato = dato;  
 this.sted = sted;  
 }  
  
 //GETTER OG SETTER FOR ID  
 public String getID() {  
 return ID;  
 }  
 public void setID(String ID) {  
 this.ID = ID;  
 }  
  
 //GETTER OG SETTER FOR FLOAT (EKSEMPEL)  
 public Float getLengde() {  
 return lengde;  
 }  
 public void setLengde(Float lengde) {  
 this.lengde = lengde;  
 }  
  
 //OPPRETTER TOSTRING-METODE:  
 @Override  
 public String toString() {  
 return "ID: "+ ID +  
 ", Kjønn: "+ kjonn +  
 ", Lengde: "+ lengde +" cm"+  
 ", Vekt: "+ vekt +" kg"+  
 ", Dato: "+ dato +  
 ", Sted: " + sted;  
 }  
  
 //OPPRETTER COMPARETO-METODEN:  
 public int compareTo(Dyr annetDyr) {  
 return this.ID.compareTo(annetDyr.getID());  
 }  
 //OPPRETTER EQUALS-METODEN:  
 public boolean equals(Dyr annetDyr) {  
 return this.ID.equals(annetDyr.getID());  
 }  
}

SUBKLASSEN GAUPE:

package KollarOblig2;  
//SUBKLASSE (DYR) GAUPE  
  
public class Gaupe extends Dyr{  
 //EGNE VARIABLER FOR DYRET GAUPE:  
 private Float tusseLengde;  
  
 //KONSTRUKTØR:  
 public Gaupe(String ID, String kjonn, Float lende, Float vekt, String dato, String sted, Float tusseLengde) {  
 super(ID, kjonn, lende, vekt, dato, sted);  
 this.tusseLengde = tusseLengde;  
 }  
   
 //GETTERE OG SETTERE  
 public Float getTusseLengde() {  
 return tusseLengde;  
 }  
 public void setTusseLengde(Float tusseLengde) {  
 this.tusseLengde = tusseLengde;  
 }  
  
 //TOSTRING  
 @Override  
 public String toString() {  
 return super.toString() +", Tusselengde: "+ tusseLengde +" cm";  
 }  
}

SUBKLASSEN HARE:

package KollarOblig2;  
//SUBKLASSE (DYR) HARE:  
  
public class Hare extends Dyr{  
 //EGNE VARIABLER FOR HARE  
 private String hareType;  
 private String harePels;  
  
 //KONSTRUKTØR:  
 public Hare(String ID, String kjonn, Float lende, Float vekt, String dato, String sted, String hareType, String harePels) {  
 super(ID, kjonn, lende, vekt, dato, sted);  
 this.hareType = hareType;  
 this.harePels = harePels;  
 }  
  
 //GETTER OG SETTERE:  
 public String getHareType() {  
 return hareType;  
 }  
 public void setHareType(String hareType) {  
 this.hareType = hareType;  
 }  
 public String getHarePels() {  
 return harePels;  
 }  
 public void setHarePels(String harePels) {  
 this.harePels = harePels;  
 }  
  
 //toString  
 @Override  
 public String toString() {  
 return super.toString() + ", Type: "+ hareType + ", Pels: "+ harePels;  
 }  
}

SUPERKLASSE GJENFANGST:

package KollarOblig2;  
//SUPERKLASSE GJENFANGST:  
  
public class Gjenfangst implements Comparable<Gjenfangst>{  
 //FELLES VARIABLER FOR DYRENE:  
 private String ID;  
 private Float nyLengde;  
 private Float nyVekt;  
 private String nyDato;  
 private String nyttSted;  
  
 //KONSTRUKTØR:  
 public Gjenfangst(String ID, Float nyLengde, Float nyVekt, String nyDato, String nyttSted) {  
 this.ID = ID;  
 this.nyLengde = nyLengde;  
 this.nyVekt = nyVekt;  
 this.nyDato = nyDato;  
 this.nyttSted = nyttSted;  
 }  
  
 //GETTERE OG SETTERE:  
 public String getID() {  
 return ID;  
 }  
  
 public void setID(String ID) { this.ID = ID; }  
  
 //TOSTRING:  
 @Override  
 public String toString() {  
 return "ID: "+ ID +  
 ", Lengde: "+ nyLengde +" cm"+  
 ", Vekt: "+ nyVekt +" kg"+  
 ", Dato: "+ nyDato +  
 ", Sted: "+ nyttSted;  
 }  
  
 //COMPARETO OG EQUALS:  
 public int compareTo(Gjenfangst annenGjenfangst) {  
 return this.ID.compareTo(annenGjenfangst.getID());  
 }  
   
 public boolean equals(Gjenfangst annenGjenfangst) {  
 return this.ID.equals(annenGjenfangst.getID());  
 }  
}

SUBKLASSE (GJENFANGST) GAUPE:

package KollarOblig2;  
  
public class NyGaupe extends Gjenfangst{  
 //EGNE VARIABLER FOR GAUPE:  
 private Float nyTusselengde;  
  
 //KONSTRUKTØR:  
 public NyGaupe(String ID, Float nyLengde, Float nyVekt, String nyDato, String nyttSted, Float nyTusselengde) {  
 super(ID, nyLengde, nyVekt, nyDato, nyttSted);  
 this.nyTusselengde = nyTusselengde;  
 }  
  
 //GETTER OG SETTER:  
 public Float getNyTusselengde() {  
 return nyTusselengde;  
 }  
  
 public void setNyTusselengde(Float nyTusselengde) {  
 this.nyTusselengde = nyTusselengde;  
 }  
  
 //TOSTRING:  
 @Override  
 public String toString() {  
 return super.toString() +", Tusselengde: "+ nyTusselengde +" cm"+ "\n";  
 }  
  
}

SUBKLASSE (GJENFANGST) HARE:

package KollarOblig2;  
  
public class NyHare extends Gjenfangst{  
 //EGNE VARIABLER FOR HARE:  
 private String nyPels;  
  
 //KONSTRUKTØR:  
 public NyHare(String ID, Float nyLengde, Float nyVekt, String nyDato, String nyttSted, String nyPels) {  
 super(ID, nyLengde, nyVekt, nyDato, nyttSted);  
 this.nyPels = nyPels;  
 }  
  
 //GETTER OG SETTER:  
 public String getNyPels() {  
 return nyPels;  
 }  
  
 public void setNyPels(String nyPels) {  
 this.nyPels = nyPels;  
 }  
  
 //TOSTRING:  
 @Override  
 public String toString() {  
 return super.toString() +", Pels: "+ nyPels +"\n";  
 }  
}

KONTROLL:

package KollarOblig2;  
//KONTROLLKLASSE:  
  
import java.util.ArrayList;  
import java.util.Collections;  
  
public class Kontroll {  
 //OPPRETTER LISTER FOR DYRENE  
 private ArrayList<Gaupe> gauper = new ArrayList<>();  
 private ArrayList<Hare> harer = new ArrayList<>();  
 private ArrayList<NyGaupe> nyeGauper = new ArrayList<>();  
 private ArrayList<NyHare> nyeHarer = new ArrayList<>();  
  
 //FUNKSJON - AUTOGENERERING AV ID OG REGISTRERING AV GAUPE:  
 public void regGaupe(String kjonn, Float lengde, Float vekt, String dato, String sted, Float tuss) {  
 String id = "G" +(gauper.size() + 1);  
 gauper.add(new Gaupe(id,kjonn,lengde,vekt,dato,sted,tuss));  
 //ALT OK  
 }  
  
 //FUNKSJON - AUTOGENERERING AV ID OG REGISTRERING AV HARE:  
 public void regHare(String kjonn,Float lengde,Float vekt,String dato,String sted,String type,String pels) {  
 String id = "H" +(harer.size() + 1);  
 harer.add(new Hare(id,kjonn,lengde,vekt,type,pels,dato,sted));  
 //ALT OK  
 }  
 //FUNKSJON - REGISTRERING AV GAUPEGJENFANGST:  
 public void regGjenGaupe(String ID, Float lengde, Float vekt, String dato, String sted, Float tuss) {  
 nyeGauper.add(new NyGaupe(ID,lengde,vekt,dato,sted,tuss));  
  
 }  
  
 //FUNKSJON - REGISTRERING AV HAREGJENFANGST:  
 public void regGjenHare(String ID, Float lengde, Float vekt, String dato, String sted, String pels) {  
 nyeHarer.add(new NyHare(ID,lengde,vekt,dato,sted,pels));  
 //ALT OK  
 }  
  
 //FUNKSJON - BINÆRT SØK I GAUPELISTE:  
 public Gaupe finnGaupeBinaert(String ID) {  
 Collections.*sort*(gauper);  
 Gaupe dummy = new Gaupe(ID,null,null,null,null,null,null);  
 int indeks = Collections.*binarySearch*(gauper, dummy);  
 if (indeks>=0) return gauper.get(indeks);  
 else return null;  
 //ALT OK  
 }  
  
 //FUNKSJON - BINÆRT SØK I HARELISTE:  
 public Hare finnHareBinaert(String ID) {  
 Collections.*sort*(harer);  
 Hare dummy = new Hare(ID,null,null,null,null,null,null,null);  
 int indeks = Collections.*binarySearch*(harer, dummy);  
 if (indeks>=0) return harer.get(indeks);  
 else return null;  
 //ALT OK  
 }  
  
 //FUNKSJON - LINJÆRT SØK FOR GJENFANGST AV GAUPE  
 public String finnGjenGaupeLinjaert(String ID) {  
 Boolean funnetGaupe = false;  
 String gaupeListe = "";  
 for (int i = 0; i< nyeGauper.size(); i++) {  
 NyGaupe id = nyeGauper.get(i);  
 if (id.getID().equals(ID)) {  
 funnetGaupe = true;  
 gaupeListe += nyeGauper.get(i);  
 }  
 }  
 if (funnetGaupe.equals(true)) return gaupeListe;  
 else return null;  
 }  
  
 //FUNKSJON - LINJÆRT SØK FOR GJENFANGST AV HARE:  
 public String finnGjenHareLinjaert(String ID) {  
 Boolean funnetHare = false;  
 String hareListe = "";  
 for (int i = 0; i< nyeHarer.size(); i++) {  
 NyHare id = nyeHarer.get(i);  
 if (id.getID().equals(ID)) {  
 funnetHare = true;  
 hareListe += nyeHarer.get(i);  
 }  
 }  
 if (funnetHare.equals(true)) return hareListe;  
 else return null;  
 }  
  
 //GETTER FOR ALLE LISTER:  
 public ArrayList<Gaupe> getGauper() {  
 return gauper;  
 }  
 public ArrayList<Hare> getHarer() {  
 return harer;  
 }  
 public ArrayList<NyGaupe> getNyeGauper() {  
 return nyeGauper;  
 }  
 public ArrayList<NyHare> getNyeHarer() {  
 return nyeHarer;  
 }  
}

GRENSESNITT:

package KollarOblig2;  
//GRENSESNITTKLASSE:  
  
import javax.swing.\*;  
import java.util.ArrayList;  
  
public class Grensesnitt {  
  
 Kontroll kontroll = new Kontroll();  
  
 // EN ARRAY MED MENYKNAPPER  
 private static final String[] *HOVEDMENY* = {"Registrer gaupe","Registrer hare",  
 "Registrer gaupegjenfangst","Registrer haregjenfangst",  
 "Finn dyr","List ut dyr","Dyr med fangst","Avslutt"};  
  
 //FUNKSJON - VISER MENYEN OG BER BRUKEREN OPPGI HVA SOM SKAL GJØRES:  
 public int lesValg() {  
 int valg = JOptionPane.*showOptionDialog*(null,"Velg hva som skal gjøres",  
 "Registrering av dyr",JOptionPane.*DEFAULT\_OPTION*,JOptionPane.*PLAIN\_MESSAGE*,  
 null,*HOVEDMENY*,*HOVEDMENY*[0]);  
 return valg;  
 }  
   
 //FUNKSJON - BRUKEREN OPPGIR DATA SOM SKAL SETTES I GAUPELISTE  
 public void lesGaupe() {  
 String kjonn = JOptionPane.*showInputDialog*("Oppgi kjønn").toUpperCase();  
 Float lengde = Float.*parseFloat*(JOptionPane.*showInputDialog*("Oppgi lende"));  
 Float vekt = Float.*parseFloat*(JOptionPane.*showInputDialog*("Oppgi vekt"));  
 Float tuss = Float.*parseFloat*(JOptionPane.*showInputDialog*("Oppgi tuss"));  
 String dato = JOptionPane.*showInputDialog*("Oppgi dato");  
 String sted = JOptionPane.*showInputDialog*("Oppgi sted").toUpperCase();  
 kontroll.regGaupe(kjonn,lengde,vekt,dato,sted,tuss);  
 }  
   
 //FUNKSJON - BRUKEREN OPPGIR DATA SOM SKAL SETTES I HARELISTE  
 public void lesHare() {  
 String kjonn = JOptionPane.*showInputDialog*("Oppgi kjønn").toUpperCase();  
 Float lengde = Float.*parseFloat*(JOptionPane.*showInputDialog*("Oppgi lende"));  
 Float vekt = Float.*parseFloat*(JOptionPane.*showInputDialog*("Oppgi vekt"));  
 String type = JOptionPane.*showInputDialog*("Oppgi type").toUpperCase();  
 String pels= JOptionPane.*showInputDialog*("Oppgi pels").toUpperCase();  
 String dato = JOptionPane.*showInputDialog*("Oppgi dato");  
 String sted = JOptionPane.*showInputDialog*("Oppgi sted").toUpperCase();  
 kontroll.regHare(kjonn,lengde,vekt,dato,sted,type,pels);  
 }  
  
 //FUNKSJON - BRUKEREN OPPGIR DATA SOM SKAL SETTES I GJENFANGST-GAUPELISTE  
 //SJEKKER OGSÅ OM DYRET FINNES FRA FØR  
 public void lesGjenfangstGaupe() {  
 String id = JOptionPane.*showInputDialog*("Oppgi id").toUpperCase();  
 Gaupe gaupe = kontroll.finnGaupeBinaert(id);  
 if (gaupe != null) {  
 Float lengde = Float.*parseFloat*(JOptionPane.*showInputDialog*("Oppgi lengde"));  
 Float vekt = Float.*parseFloat*(JOptionPane.*showInputDialog*("Oppgi vekt"));  
 Float tuss = Float.*parseFloat*(JOptionPane.*showInputDialog*("Oppgi tusselengde"));  
 String dato = JOptionPane.*showInputDialog*("Oppgi dato");  
 String sted = JOptionPane.*showInputDialog*("Oppgi sted").toUpperCase();  
 kontroll.regGjenGaupe(id, lengde, vekt, dato, sted, tuss);  
 }else JOptionPane.*showMessageDialog*(null, "Finner ikke gaupen med denne ID'en");  
 }  
  
 //FUNKSJON - BRUKEREN OPPGIR DATA SOM SKAL SETTES I GJENFANGST-HARELISTE  
 //SJEKKER OGSÅ OM DYRET FINNES FRA FØR  
 public void lesGjenfangstHare() {  
 String id = JOptionPane.*showInputDialog*("Oppgi id").toUpperCase();  
 Hare hare = kontroll.finnHareBinaert(id);  
 if (hare != null) {  
 Float lengde = Float.*parseFloat*(JOptionPane.*showInputDialog*("Oppgi lende"));  
 Float vekt = Float.*parseFloat*(JOptionPane.*showInputDialog*("Oppgi vekt"));  
 String pels= JOptionPane.*showInputDialog*("Oppgi pels").toUpperCase();  
 String dato = JOptionPane.*showInputDialog*("Oppgi dato");  
 String sted = JOptionPane.*showInputDialog*("Oppgi sted").toUpperCase();  
 kontroll.regGjenHare(id, lengde, vekt, dato, sted, pels);  
 } else JOptionPane.*showMessageDialog*(null,"Finner ikke haren med denne ID'en");  
 }  
   
 //FUNKSJON - OPPGI DYRETS ID, PROGRAMMET SKRIVER UT DYRETS DATA OG GJENFANGSTDATA  
 public void lesFinnDyr() {  
 String id = JOptionPane.*showInputDialog*("Oppgi ID").toUpperCase();  
 Gaupe gaupe= kontroll.finnGaupeBinaert(id);  
 String gaupe1 = kontroll.finnGjenGaupeLinjaert(id);  
 Hare hare = kontroll.finnHareBinaert(id);  
 String hare1 = kontroll.finnGjenHareLinjaert(id);  
  
 if (gaupe !=null) System.*out*.println("Første registrering av gaupe:"+"\n"+gaupe.toString());  
  
 if (gaupe1 !=null) System.*out*.println("Gjenfangst data for gaupe:"+"\n"+gaupe1.toString());  
  
 if (hare !=null) System.*out*.println("Første registrering av hare:"+"\n"+hare.toString());  
  
 if (hare1 !=null) System.*out*.println("Gjenfangst data for hare:"+"\n"+hare1.toString());  
 }  
  
 //FUNKSJON - LIST DYR MED DATA OG GJENFANGSTDATA:  
 public void lesListDyr() {  
 ArrayList<Gaupe> gauper = kontroll.getGauper();  
 ArrayList<NyGaupe> nyeGauper = kontroll.getNyeGauper();  
 ArrayList<Hare> harer = kontroll.getHarer();  
 ArrayList<NyHare> nyeHarer = kontroll.getNyeHarer();  
 String gaupe = "";  
 String gaupe1 = "";  
 String hare = "";  
 String hare1 = "";  
 for (Gaupe g : gauper) {  
 gaupe+= g.toString() + "\n";  
 }  
 for (NyGaupe ng : nyeGauper) {  
 gaupe1+= ng.toString();  
 }  
 for (Hare h : harer) {  
 hare+= h.toString() + "\n";  
 }  
 for (NyHare nh : nyeHarer) {  
 hare1+= nh.toString();  
 }  
 System.*out*.println("Første registrering av gaupe:" + "\n" + gaupe + "Gjenfangst:" + "\n" + gaupe1 + "\n" +  
 "Første registrering av hare:" + "\n" + hare + "Gjenfangst:" + "\n" + hare1);  
 }  
  
  
 //FUNKSJON - GJENFANGSTDATA FOR ALLE DYR  
 //TILBAKEMELDING DERSOM DYRET IKKE HAR GJENFANGST:  
 public void lesDyrMedFangst() {  
 ArrayList<NyGaupe> gaupeGjenfangst = kontroll.getNyeGauper();  
 ArrayList<NyHare> hareGjenfangst = kontroll.getNyeHarer();  
 String utGjenGaupe = "";  
 String utGjenHare = "";  
 for (NyGaupe nyGaupe : gaupeGjenfangst) {  
 utGjenGaupe+=nyGaupe.toString();  
 }  
 for (NyHare nyHare : hareGjenfangst) {  
 utGjenHare+=nyHare.toString();  
 }  
 if (utGjenGaupe.isEmpty()) System.*out*.println("Ingen gjenfangstdata registrert for Gaupe");  
 else System.*out*.println("Gjenfangstdata for Gaupe:"+"\n"+utGjenGaupe);  
 if (utGjenHare.isEmpty()) System.*out*.println("Ingen gjenfangstdata for Hare");  
 else System.*out*.println("Gjenfangstdata for Hare:"+"\n"+utGjenHare);  
 }  
  
 //FUNKSJON - MENYEN KJØRES SÅ LENGE MAN IKKE TRYKKER AVSLUTT:  
 public void start() {  
  
 boolean fortsett = true;  
  
 while (fortsett) {  
 int valg = lesValg();  
  
 switch (valg) {  
 case 0 :  
 lesGaupe();  
 break;  
 case 1 :  
 lesHare();  
 break;  
 case 2 :  
 lesGjenfangstGaupe();  
 break;  
 case 3 :  
 lesGjenfangstHare();  
 break;  
 case 4 :  
 lesFinnDyr();  
 break;  
 case 5 :  
 lesListDyr();  
 break;  
 case 6 :  
 lesDyrMedFangst();  
 break;  
 default: fortsett = false;  
 }  
 }  
 }  
}

MAIN:

package KollarOblig2;  
//MAINKLASSE:  
  
public class Main {  
 //FUNKSJON - STARTER PROGRAMMET  
   
 public static void main(String[] args) {  
 Grensesnitt programStart = new Grensesnitt();  
  
 //HENTER INN start()-funksjonen FRA Grensesnitt  
 programStart.start();  
 }  
}