Løsningsforslag – uke 7

(fra gjennomgang i gruppetimen)

1.

```
public class Person implements Comparable<Person>{
  private String fornavn;
  private String etternavn;
  private int alder;

  public Person(String fn, String en, int a){
    fornavn = fn;
    etternavn = en;
    alder = a;
  }

  @Override
  public String toString(){
    return fornavn;
  }

  public int hentAlder(){
    return alder;
  }

  @Override
  public int compareTo(Person p){
    if(alder > p.hentAlder()) return 1;
    if(alder < p.hentAlder()) return -1;
    if(fornavn.compareTo(p.toString()) > 0) return 1;
    if(fornavn.compareTo(p.toString()) < 0) return -1;
    return 0;

    /* Alternativ måte:
    if(alder == p.hentAlder()){
      return toString().compareTo(p.toString());
    }
    return alder-p.hentAlder();
    */
  }
}
```

2.

FIFOListe.java

```
class FIFOListe<T> implements Iterable<T> {

  private Node hode;
  private Node hale;

  private class Node {
    Node neste;
    T ting;

    Node(T ting) {
      this.ting = ting;
    }
  }

  public int storrelse() {
    int teller = 0;
    for (Node tmp = hode; tmp != null; tmp = tmp.neste) {
      teller++;
    }
    return teller;
  }

  public boolean erTom() {
    return hode == null;
  }

  public void leggTil(T t) {
    Node nyNode = new Node(t);
    if (erTom()) {
      hode = nyNode;
    }
    else {
      hale.neste = nyNode;
    }
    hale = nyNode;
  }

  public void fjernAlt() {
    hode = null;
    hale = null;
  }

  public T pop() {
    Node returnode = hode;
    hode = hode.neste;
    return returnode.ting;
  }

  public void skrivUtListe() {
    Node tmp = hode;
    System.out.println("[");
    while (tmp != null) {
      System.out.println("\t" + tmp.ting);
      tmp = tmp.neste;
    }
    System.out.println("]");
  }

  public Iterator<T> iterator() {
    return new FIFOIterator();
  }

  private class FIFOIterator implements Iterator<T> {

    Node denne = hode;

    @Override
    public boolean hasNext() {
      return denne != null;
    }

    @Override
    public T next() {
      if (denne == null) throw new NoSuchElementException("next")
      Node tmp = denne;
      denne = denne.neste;
      return tmp.ting;
    }
  }
}
```

TestFIFO.java

```
class TestFIFO {
  public static void main(String[] args) {
    FIFOListe<String> liste = new FIFOListe<>();

    liste.leggTil("Hei");
    liste.leggTil("på");
    liste.leggTil("deg");
    liste.leggTil("!");

    System.out.println("\nBruker skrivUtListe() ... ");
    liste.skrivUtListe();

    System.out.println("\nBruker iterator ... ");
    for (String string : liste) {
      System.out.print(string + " ");
    }
    System.out.println();

    System.out.println("\nBruker pop() ... ");
    System.out.print(liste.pop() + " ");
    System.out.print(liste.pop() + " ");
    System.out.print(liste.pop());
    System.out.println(liste.pop());
  }
}
```