Aufgabe 8-2d

```
// while-Schleife

int n = 1;
double x = 0;
double s = 1.0 / (n * n);
x = x + s;
n++;

while (s > 0.01) {
    s = 1.0 / (n * n);
    x = x + s;
    n++;
}

// for-Schleife

int n = 1;
double x = 0;
double s = 1.0 / (n * n);
x = x + s;
n++;

for (; s > 0.01; ) {
    s = 1.0 / (n * n);
    x = x + s;
    n++;
}
```

Aufgabe 8-3a

```
public static double exp(double x, int n) {
    double result = x;
    while (n > 1) {
        result = result * x;
        n--;
    }
    return result;
}
```

Aufgabe 8-3b

```
public static float fakultaet(int n) {
    float result = n--;
    while (n > 0) {
        result = result * n;
        n--;
    }
    return result;
}
```

Aufgabe 8-3c

```
public static boolean istPrim(int n) {
    // moegliche Teiler liegen im Bereich 2 bis n/2 (abgerundet)
    for (int zaehler = 2; zaehler <= n / 2; zaehler++) {
        if (n % zaehler == 0) {
            return false;
        }
    }
    return true;
}
```

Aufgabe 8-3d

```
public static int ggt(int m, int n) {
    while (m != n) {
        if (m > n) {
            m = m - n;
        } else {
            n = n - m;
        }
    }
    return m;
}
```

Aufgabe 8-3e

```
public static char getValue(int i) {
    switch (i) {
       case 0:  return '0';
       case 1:  return '1';
       case 2:  return '2';
       case 3:  return '3';
       case 4:  return '4';
       case 5:  return '5';
       case 6:  return '6';
       case 7:  return '7';
       case 8:  return '8';
       case 9:  return '9';
       case 10: return 'A';
       case 11: return 'B';
       case 12: return 'C';
       case 13: return 'D';
       case 14: return 'E';
       case 15: return 'F';
       default: return '%';
    }
}

public static String rebase(int n, int p) {
    if (n == 0) {
        return "0";
    }
    String result = "";
    while (n > 0) {
        result = getValue(n % p) + result;
        n = n / p;
    }
    return result;
}
```

Aufgabe 8-4

```
public class Wuerfeln {

    public static void main(String[] args){
        muenzwurf(3);
        System.out.println();
        wuerfeln(2,5);
    }

    public static int exp(int b, int e){
        int result = 1;
        for (int i = 0; i < e; i++){
            result *= b;
        }
        return result;
    }

    public static void muenzwurf(int n){
        for (int i = 0; i < Math.pow(2, n); i++){
            map2(i,n);
        }
    }

    public static void wuerfeln(int n, int k ){
        for (int i = 0; i < Math.pow(k, n); i++){
            map(i, k, n);
        }
    }

    public static void map(int n, int k, int length){
        for (int i = 0; i < length; i++){
            int t = n%k;
            n = n / k;
            System.out.print(t+1);
            if(i == length-1){
                System.out.println();
            } else {
                System.out.print(",");
            }
        }
    }

    public static void map2(int z, int n){
        for (int i = 0; i < n; i++){
            int t = z%2;
            z = z / 2;
            if (t == 0){
                System.out.print("K");
            } else {
                System.out.print("Z");
            }
            if (i == n-1){
                System.out.println();
            } else {
                System.out.print(",");
            }
        }
    }
}
```