

Lösung:

1.

a)

```
public class JUGENDLICHER{
    private int alter;
    private String name;
    private boolean raucher;
    private int punkte;
    private ZUFALL z;
```

b) JUGENDLICHER(String n){
 alter = 10;
 name = n;
 raucher = false;
 punkte = 0;
 z = new ZUFALL();
}

c) int Summe(int a, int b){
 int ergebnis = a + b + z.GanzzahlGeben(3);
 System.out.println(a + " + " + b + " = " + ergebnis);
 return ergebnis;
}

d) int Produkt(int a, int b){
 int ergebnis = a * b + z.GanzzahlGeben(6);
 System.out.println(a + " * " + b + " = " + ergebnis);
 return ergebnis;
}

e) void punkteErhöhen(){
 punkte = punkte + 1;
}

```
int punkteGeben(){
    return punkte;
}
```

```
void punkteAufNull(){
    punkte = 0;
}
```

2.

c) void rauchenAngewöhnen(){
 raucher = true;
}
}

```
public class SCHUELER extends JUGENDLICHER{
```

a) SCHUELER(String n){
 super(n);
 //raucher = true; geht nicht, Fehlermeldung, weil das Attribut
 gekapselt ist, deshalb:
 rauchenAngewöhnen();
}

b) @Override int Summe(int a, int b){
 int ergebnis = a + b;
 System.out.println(a + " + " + b + " = " + ergebnis);
 return ergebnis;
}
}

3.

```
public class LEHRER{
a) private SCHUELER s1;
    private JUGENDLICHER j1;
    private ZUFALL z;

    public LEHRER(){
        s1 = new SCHUELER("Otto");
        j1 = new JUGENDLICHER("Fritz");
        z = new ZUFALL();
    }

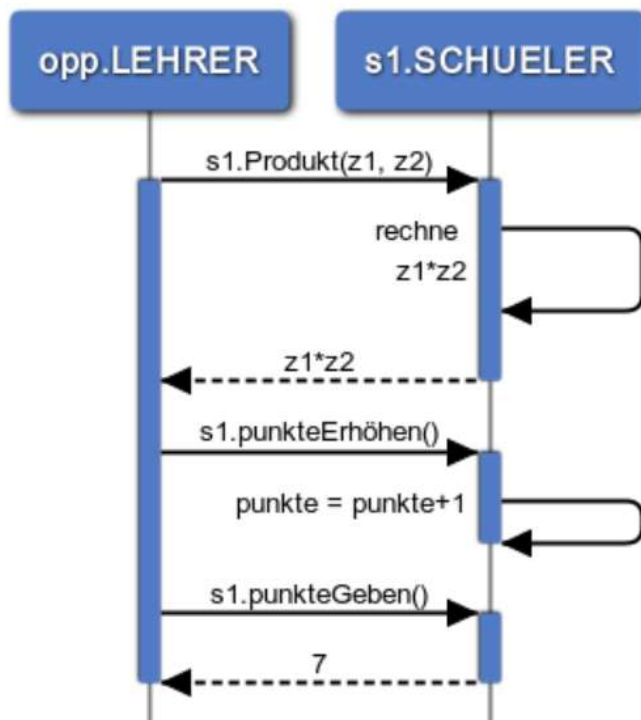
b) void abfragen(JUGENDLICHER zf){
    zf.punkteAufNull();
    for (int i=1; i<=10; i++){
        int v1 = z.GanzzahlGeben(21);
        int v2 = z.GanzzahlGeben(21);
        if (zf.Summe(v1,v2) == v1+v2){
            zf.punkteErhöhen();
        }
    }
    System.out.println("Deine Punkte: " + zf.punkteGeben());
}

c) void testen(){
    abfragen(s1);
    abfragen(j1);
    rauchenBeibringen(j1);
    noteGeben(j1);
}

5. void rauchenBeibringen(JUGENDLICHER j){
    //j.raucher = true;
6.    j.rauchenAngewöhnen();
    }

7. void noteGeben(JUGENDLICHER j){
    int a = j.punkteGeben();
    switch(a){
        case 10:{
            System.out.println("Note 1");
        }break;
        case 9:{
            System.out.println("Note 2");
        }break;
        case 8: case 7:{
            System.out.println("Note 3");
        }break;
        case 6: case 5:{
            System.out.println("Note 4");
        }break;
        case 4: case 3:{
            System.out.println("Note 5");
        }break;
        case 2: case 1:{
            System.out.println("Note 6");
        }break;
        default:{
            System.out.println("Du sprengst die Skala!");
        }break;
    }
}
}
```

4.



```
opp.LEHRER -> +s1.SCHUELER: s1.Produkt(z1, z2)
activate opp.LEHRER
s1.SCHUELER -> s1.SCHUELER: rechne \n z1*z2
s1.SCHUELER --> -opp.LEHRER: z1*z2
opp.LEHRER -> +s1.SCHUELER: s1.punkteErhöhen()
s1.SCHUELER -> -s1.SCHUELER: punkte = punkte+1
opp.LEHRER -> +s1.SCHUELER: s1.punkteGeben()
s1.SCHUELER --> -opp.LEHRER: 7
deactivate opp.LEHRER
```

5. Fehlermeldung „raucher has private access in JUGENDLICHER“ weil das Attribut gekapselt ist. Zugriff auf das Attribut ist nur über einen Methodenaufruf möglich.