Zu 1:

**Arbeitnehmer**

private String name;

private int erfahrungslevel;

private boolean extern;

private boolean intern;

private boolean urlaub;

getUrlaub();

setUrlaub();

get Erfahrungslevel (hier kommt noch die Prüfung ob der Wert kleiner 6 ist);

getName();

setName(hier kommt noch die Prüfung rein ob der Name länger als 255 ist rein.);

Die Klasse zeigt wie ein Objekt aussehen soll/kann. Sie ist der Bauplan eines Objektes.

Sie hat die Fähigkeit Methoden auszuführen. So kann Sie zu zum Beispiel durch eine

Methode prüfen ob der MA ein externer oder interner ist und ob er sich im Urlaub befindet.

Eine 2 Methode kann prüfen ob der MA ein Namen Erfahrungslevel besitzt.

Zu 2:

Ich nutze für diese Aufgabe Konstruktoren einsetzen.

Public Arbeitnehmer()

{ name = „“; erfahrungslevel = 0; intern = true; extern = false; urlaub = false; }

Public Arbeitnehmer (String name, int erfahrungslevel) {

this.name = name; this.erfahrungslevel = erfahrungslevel; this.intern = intern;

this.extern = extern; this.urlaub = urlaub; }

Zu 3:

1. Er bekommt die Sichtbarkeit private und keine getter und setter Methode.
2. Er bekommt die Sichtbarkeit private und die getter Methode aber keine setter Methode.
3. Er bekommt die Sichtbarkeit private und die getter und setter Methode.
4. Er bekommt die Sichtbarkeit private und die getter und setter Methode. In der setter Methode muss aber noch die Prüfung rein ob Name >= 255 Zeichen ist mit String.length.

**Zu Tester, Programmierer und GUI-Entwickler**

**Arbeitnehmer**

Die Klassen erben alle Eigenschaften und Methoden der Superklasse.

**Tester**  **Programmierer GUI-Entwickler**

Testen; programmieren; huebschMachen;

Int schlaf; int zufriedenheit; int koffeingehalt;

getTesten(); getProgrammieren(); getHuebschmachen();

setTesten(); setProgrammieren(); setHuebschmachen();

getSchlaf(); getZufriedenheit(); setKoffeingehalt();

setSchlaf(); setZufriedenheit(); setKoffeingehalt();

zu2:

Ja ist möglich. Da die Beziehung zur Superklasse (Arbeitnehmer) besteht und dort die Berechnung zur Verfügbarkeit erfolgt. Diese wird den Subklassen vererbt bzw. sind diese für Sie zugänglich.

Void verfügbar();

Zu3:

Ich muss dafür einen Konstruktor erstellen der 2 Parameter erwartet.

Public Programmierer (String Name, int Erfahrungslevel) { }

Sichtbarkeit Name der Klasse Parameter

Zu4:

Public boolean testen (Arbeitspaket Testpaket) { }

Sichtbarkeit Rückgabewert Name der Klasse Parameter (besteht aus Typ und Name)

Hier können viele rein kann da Ulrike glauben.

**Arbeitspaket**

Zu 1:

**Arbeitspaket**

private String typ;

private String soll;

private String ist;

private String beschreibung;

boolean geprueft;

getTyp(); set Typ();

getSoll(); setSoll();

getIst(); setIst();

getBezeichnung(); setBezeichnung();

boolean gefuellt();

Public Arbeitspaket() {

beschreibung = „“;

typ = „“;

soll = „“;

ist = „“;

}

Public Arbeitspaket (String soll, String typ) {

this. beschreibung = beschreibung;

this. typ = typ;

this. soll = soll;

this. ist = ist

}

Boolean gefuellt () {

If (getIst() == „“) return false;

} else {

Return true;

}

}

**Projekt**

Zu 1:

**Projekt**

Private String Name;

private int AnzahlAPGUI;

private int AnzahlAPProgrammieren;

getName(); set Name();

getAnzahlAPGUI(); set AnzahlAPGUI ();

getAnzahlAPProgrammieren(); setAnzahlAPProgrammieren();

boolean gefuellt(); getAnzahlAPTester ()

getAnzahlAPGesamt();

public int **getAnzahlAPTester** ()

{

return (this. AnzahlAPGUI + this. AnzahlAPProgrammieren);

}

public int getAnzahlAPGesamt ()

{

return (this. AnzahlAPGUI + this. AnzahlAPProgrammieren)+ this. **getAnzahlAPTester()**;

}

Zu2:

Die Arbeitspakete würde ich in ein Array packen. Die Anzahl der der Arbeitspakete ist die Größe des Array. Damit kann ich dann den Fortschritt errechnen.

Zu3:

Public Projekt(String Name, int APProgrammierung, int APGUI) {

this.name = name;

this. APProgrammierung = APProgrammierung;

this. APGUI = APGUI;

this. APTesten = APTesten;

this.Gesamtfortschritt = Gesamtfortschritt;

this.AnzahlAPGesamt = AnzahlAPGesamt;

}

Zu 4:

Ich würde die Methode rechneFortschritt dafür nutzen. Da ich die Anzahl der Arbeitspakete in einem Array habe und dort auch den Fortschritt der einzelenen Pakete, sollte es möglich sein den Fortschritt für die einzelnen Arbeitspakete zuerhalten. Diese würde ich dann addieren und könnte so den Gesamtfortschritt errechnen.

Arbeitspakete für Programmierer sind 5 davon sind 3 fertig also (3/5)\*100 = 60 %

Arbeitspakete für GUI-Entwickler sind 10 davon sind 3 fertig also (3/10)\*100 = 33 %

Arbeitspakete für Tester sind dann 15 davon sind 8 fertig also (8/15)\*100 = 53 %

Zu 5:

Ja ich kann die Methode rechneFortschritt für alle 4 Fälle nutzen

Gesamt sind es dann 30 davon sind 12 fertig also (12/30)\*100 = 40 %

Damit ist der Gesamtfortschritt 40 %.