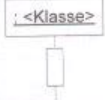
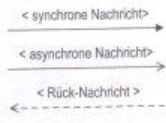
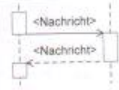

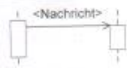
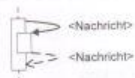
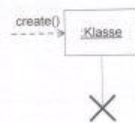


Die X&Y GmbH verwendet zur Konstruktion und Dokumentation von Software und IT-Systemen die Unified Modeling Language (UML).

1. Der Ablauf einer Bestellung im Online-Shop der PixelPic AG soll in einem UML-Sequenzdiagramm dargestellt werden. (18 P)

Der Ablauf wird wie folgt beschrieben:

- Kunde ruft die Startseite mit dem Angebot des Shops auf
- Shop zeigt Kunden die Startseite mit den zur Wahl stehenden Produkten (Foto, Poster).
- Kunde teilt Shop Produktwahl mit.
- Shop erstellt Produktvorlagen-Objekt; Aufruf der Funktion *create()* der entsprechenden Produktvorlagen-Klasse
- Shop zeigt dem Kunden auf HTML-Seite die leere Produktvorlage und die Mitteilung, dass das Bild hochgeladen werden kann.
- Kunde lädt Bild hoch.
- Shop übergibt Bild an Produktvorlage; Aufruf der Funktion *bild()* des Produktvorlagen-Objekts.
- Produktvorlage meldet "Bild ist eingefügt".
- Shop zeigt dem Kunden HTML-Seite mit fertiger Produktvorlage und fordert Kunden zur Eingabe der Adress- und Zahlungsdaten auf.
- Kunde übermittelt Daten.
- Shop ruft die eigene Funktion *pruefung()* zur Prüfung der Daten auf.
- Funktion *pruefung()* gibt "Daten o.k." zurück. Der Fall "Daten nicht o.k." wird nicht betrachtet.
- Shop teilt Kunden mit, dass er zur Annahme des Auftrags bereit ist.
- Kunde erteilt Auftrag
- Shop erstellt ein Auftragsbestätigungs-Objekt mit dem Aufruf der Funktion *create()* der Klasse *Auftragsbestaetigung*.
- Shop verschickt Auftragsbestätigung per E-Mail an den Kunden.

Notation	Beschreibung
	Objekt, Lebenslinie, Aktivierung Objekt: Rechteck, Beschriftung: : <Klasse> Lebenslinie: senkrechte, gestrichelte Linie; Zeit, in der das Objekt existiert Aktivierung: schmales Rechteck; Zeit, in der Objekt aktiv ist, z. B. Ausführung einer Methode
	Nachricht Objekte kommunizieren über Nachrichten, die als Pfeile zwischen den Aktivierungen eingezeichnet werden. - Beschriftung: – bei Methodenaufwurf mit der Bezeichnung der Methode, z. B. create() – sonst Inhalt der Nachricht, z. B. „zeige Startseite“
	sequenzielle Nachricht Der Sender erwartet vom Empfänger seiner Nachricht eine Antwort und kann erst wieder im Prozess aktiv werden, wenn die Antwort vorliegt.
	synchrone Nachricht Der Sender erwartet vom Empfänger seiner Nachricht eine Antwort.
	asynchrone Nachricht Der Sender erwartet vom Empfänger seiner Nachricht keine Antwort.
	Selbstdelegation Ein Objekt ruft eine Methode auf, die es selbst implementiert. Das Objekt kann ggf. erst fortfahren, wenn die aufgerufene Methode die Verarbeitung beendet und eine Antwort gesendet hat.
	Objekt erzeugen Richtung: von Quelle zum Kopf des neuen Objekts Darstellung: gestrichelte Linie, offener Pfeil Objekt löschen Löschen eines Objektes wird durch ein X auf der Lebenslinie markiert.