b) Process-Interaction Ansatz: Realisierung aus kunden-orientierter Sicht

Definition eines Kundenobjekts mit Attribut k
(Attribute zur Berechnung von Statistiken, wie z.B. Ankunftszeit des Kunden etc.)

```
typedef struct // Objekt Kunde
{
int k; // Anzahl Waren, die der Kunde auswaehlt
int id; // Kunden ID zur eindeutigen Identifizierung
} Kunde;
```

Hauptprogramm der Simulation

```
t = 0.0; // Simulationszeit
freie_koerbe = M; // am Anfang sind alle Koerbe frei
freie_kassen = N; // am Anfang sind alle Kassen frei
start_process(Umwelt); // ein Umweltprozess
pause_for(Simulationslaenge); // Simulationsende planen
```

Umweltprozess

```
for(;;) // Endlossschleife
{
  pause_for(ziehe_zz(EXP,λ)); // Zwischenankunftszeit
  start_process(Kunde); // Ankunft eines neuen Kunden
}
```

Kundenprozess

```
Kunde* K = new Kunde; // erzeuge neuen Kunden K
K->k = ziehe_zz(UNI,(a,b)); // Anzahl Waren auswuerfeln
K->id = PROCESS_ID; // ID zur Identifizierung

enqueue(koerbe,K); // an Koerben anstellen
wait_until ( (K == first(koerben)) && (freie_koerbe > 0) );
freie_koerbe = freie_koerbe - 1;
dequeue(koerbe);
pause_for(K->k * T_s); // Kunde sucht Waren aus

enqueue(kassen,K); // an Kassen anstellen
wait_until ( (K == first(kassen)) && (freie_kassen > 0) );
freie_kassen = freie_kassen - 1;
dequeue(kassen);
pause_for(K->k * T_p); // Kunde wird an der Kasse bedient

freie_kassen = freie_kassen - 1;// Bezahlen beendet
freie_koerbe = freie_koerbe - 1;// gebe Korb zurueck

delete(K);// Kundenobjekt loeschen
```