

Berufsfachschule Wedel gGmbH
Abschlussprüfung für Assistenten im SoSe 2021
Fach Systeme der Softwaretechnik analysieren und gestalten
2. Vorschlag

Hilfsmittel: Taschenrechner ohne CAS-Funktionalität

Dauer: 180 Minuten - **Prüfer:** Birger Wolter

Das BFS-Team wünscht viel Erfolg

Aufgabe 1

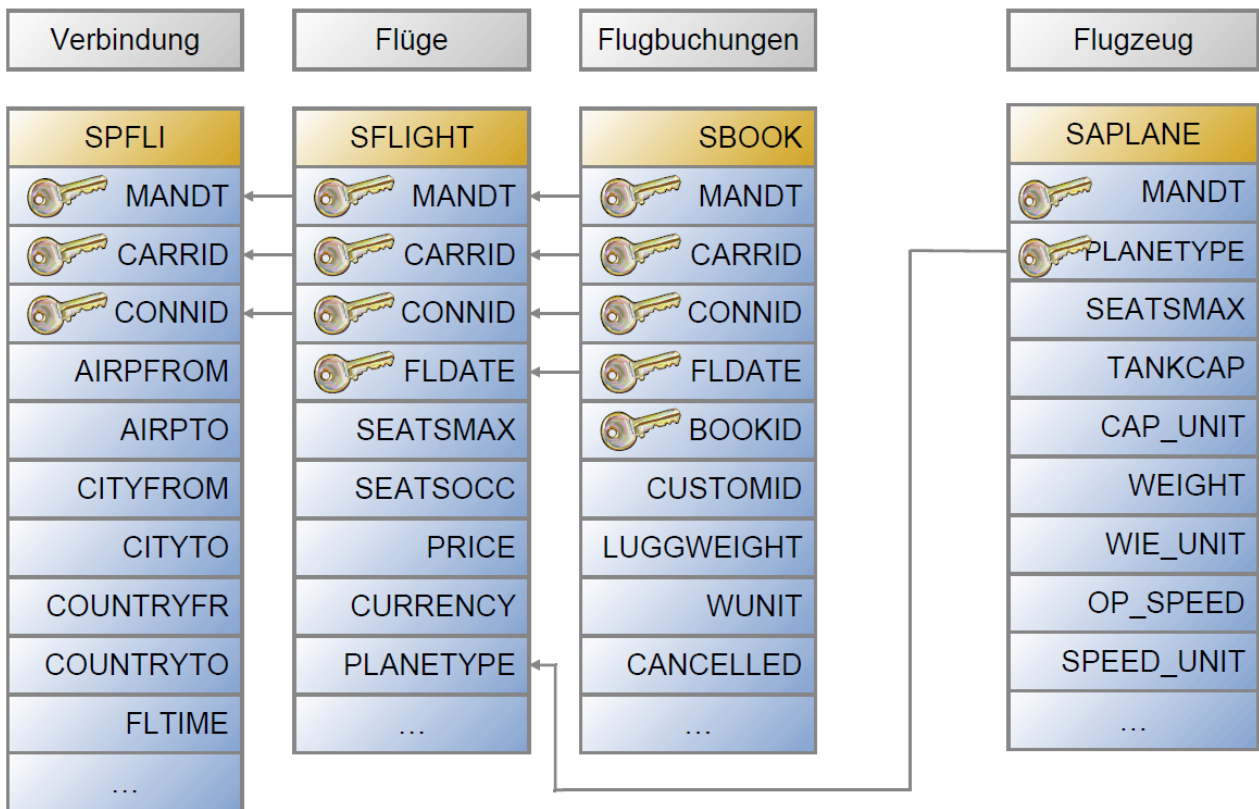
Erläutere stichwortartig mindestens drei signifikante Vorteile von Standardsoftware:

Erläutere stichwortartig mindestens drei signifikante Vorteile von Individualsoftware:

Ordne die SAP-Software begründet einer der beiden Arten zu:

Aufgabe 2

Entwickle eine Business Server Page (BSP), welche auf dem Inhalt der Datenbanktabelle SAPLANE basierend (siehe nachfolgenden Ausschnitt des SAP-Flugdatenmodells) für die drei Flugzeugtypen (PLANETYPE) mit der grössten, zweitgrössten und drittgrössten Sitzplatzanzahl (SEATSMAX) jeweils den Flugzeugtyp, die Sitzplatzanzahl, die Reisegeschwindigkeit (OP_SPEED) und die Masseinheit der Reisegeschwindigkeit (SPEED_UNIT) im HTML-Format ausgibt. Es darf davon ausgegangen werden, dass in der Datenbanktabelle keine Flugzeugtypen mit gleicher Sitzplatzanzahl vorhanden sind.



Aufgabe 2 (Fort.)

Aufgabe 3

Gebe die exakte Ausgabe des nachfolgenden Reports an, und begründe die genannte Lösung:

```
REPORT zz_objects.

CLASS train DEFINITION.
  PUBLIC SECTION.
    METHODS: setter IMPORTING
              imp_name TYPE string,
              getter.
  PRIVATE SECTION.
    DATA: name TYPE string.
ENDCLASS.

CLASS train IMPLEMENTATION.
  METHOD setter.
    name = imp_name.
  ENDMETHOD.
  METHOD getter.
    WRITE: / name.
  ENDMETHOD.
ENDCLASS.

DATA: name    TYPE string VALUE '0',
      train   TYPE REF TO train,
      trains  TYPE TABLE OF REF TO train.

START-OF-SELECTION.

  WHILE strlen( name ) < 10.
    CREATE OBJECT train.
    CONCATENATE 'X' name INTO name SEPARATED BY '-'.
    train->setter( imp_name = name ).
    INSERT train INTO trains INDEX 1.
  ENDWHILE.
  LOOP AT trains INTO train.
    train->getter( ).
  ENDLOOP.
```

Aufgabe 4

Erläutere mit eigenen Worten stichwortartig möglichst exakt sowohl die drei BSP-Bestandteile als auch den Report ZZ_OBJECTS.

Layout eingabe.htm

```
<%@page language="abap"%>
<HTML>
  <HEAD>
    <TITLE>Selektionskriterien SPFLI</TITLE>
  </HEAD>
  <BODY>
    <FORM NAME="Formular">
      <TABLE>
        <TR>
          <TD>Fluggesellschaft:</TD>
          <TD><INPUT TYPE="text" NAME="param_carrid"></TD>
        </TR><TR>
          <TD>Verbindung: </TD>
          <TD><INPUT TYPE="text" NAME="param_connid"></TD>
        </TR><TR>
          <TD>Testmodus:</TD>
          <TD><INPUT TYPE="checkbox" NAME="param_modus" value="X"></TD>
        </TR><TR>
          <TD COLSPAN="2">&nbsp;</TD>
        </TR><TR>
          <TD COLSPAN="2"><INPUT TYPE="submit" NAME="onInputProcessing(event_1)"
            VALUE="Flugdaten anzeigen"></TD>
        </TR>
      </TABLE>
    </FORM>
  </BODY>
</HTML>
```

Eventhandler OnInputProcessing eingabe.htm

```
CASE event_id.
  WHEN 'event_1'.
    navigation->set_parameter( 'param_carrid' ).
    navigation->set_parameter( 'param_connid' ).
    navigation->set_parameter( 'param_modus' ).
    navigation->goto_page( 'ausgabe.htm' ).
ENDCASE.
```

Aufgabe 4 (Fort.)

Layout ausgabe.htm

```
<%@page language="abap" %>
<HTML>
  <HEAD>
    <TITLE>Selektionsergebnis SPFLI</TITLE>
  <%
    DATA: it_fluege TYPE TABLE OF spfli,
           wa_flug TYPE spfli.
    IF param_carrid = '' AND param_connid = ''.
      SELECT * FROM spfli INTO TABLE it_fluege.
    ELSEIF param_carrid = ''.
      SELECT * FROM spfli INTO TABLE it_fluege WHERE connid = param_connid.
    ELSEIF param_connid = ''.
      SELECT * FROM SPFLI INTO TABLE it_fluege WHERE carrid = param_carrid.
    ELSE.
      SELECT * FROM spfli INTO TABLE it_fluege WHERE carrid = param_carrid
              AND connid = param_connid.

    ENDIF.
  %>
</HEAD>
<BODY>
  <H2>Ausgabe von Daten aus SPFLI</H2>
  <TABLE BORDER="1">
    <TR>
      <TH>Fluggesellschaft</TH>
      <TH>Verbindung</TH>
      <TH>Startflughafen</TH>
      <TH>Zielflughafen</TH>
      <TH>Abflugszeit</TH>
      <TH>Ankunftszeit</TH>
    </TR>
    <%
      LOOP AT it_fluege INTO wa_flug.
    %>
    <TR>
      <TD><%= wa_flug-carrid %></TD>
      <TD><%= wa_flug-connid %></TD>
      <TD><%= wa_flug-airpfrom %></TD>
      <TD><%= wa_flug-airpto %></TD>
      <TD><%= wa_flug-deptime %></TD>
      <TD><%= wa_flug-aritime %></TD>
    </TR>
    <%
      ENDLOOP.
    %>
  </TABLE>
  <%
    IF param_modus = 'X'.
  %>
  <H3>Testmodus</H3>
  <%
    ELSE.
  %>
  <H3>Produktivmodus</H3>
  <%
    ENDIF.
  %>
</BODY>
</HTML>
```

Aufgabe 4 (Fort.)

Report ZZ_OBJECTS

```
REPORT ZZ_OBJECTS.

CLASS lcl_airplane DEFINITION.

    PUBLIC SECTION.
        METHODS: attribute_setzen IMPORTING
            im_name TYPE string
            im_planetype TYPE string,
            attribute_anzeigen.

    PRIVATE SECTION.
        DATA: name TYPE string,
            planetype TYPE string.

ENDCLASS.

CLASS lcl_airplane IMPLEMENTATION.

    METHOD attribute_setzen.
        name = im_name.
        planetype = im_planetype.
    ENDMETHOD.

    METHOD attribute_anzeigen.
        WRITE: / 'Name des Flugzeugs: ', name,
            / 'Typ des Flugzeugs: ', planetype.
    ENDMETHOD.

ENDCLASS.

DATA: r_plane TYPE REF TO lcl_airplane,
    it_plane_list TYPE TABLE OF REF TO lcl_airplane.

START-OF-SELECTION.

CREATE OBJECT r_plane.
r_plane->attribute_setzen( im_name = 'München'
    im_planetype = 'Boeing 737' ).
APPEND r_plane TO it_plane_list.

CREATE OBJECT r_plane.
r_plane->attribute_setzen( im_name = 'Hamburg'
    im_planetype = 'Airbus 380' ).
APPEND r_plane TO it_plane_list.

LOOP AT it_plane_list INTO r_plane.
    r_plane->attribute_anzeigen( ).
ENDLOOP.
```

Aufgabe 4 (Fort.)

Aufgabe 4 (Fort.)