

بنام خدا

آموزش برنامه نویسی و توسعه SAP

(پیش نمایش...!)

SAP ABAP

Advanced Business Application Programming

تهیه کننده

مهندس امیر پهلوان صادق

نسخه: ۱,۱

Beyamooz.com

آموزشگاه بیاموز

مقدمه

کتاب الکترونیکی که هم اکنون پیش روی خود دارید مطالب گردآوری شده چندین سایت خارجی و تجارب بنده حین کار می باشد. بهتر بود مطالب در قالب فایل در اختیار دوستان و علاقه مندان قرار بگیرد تا از آن استفاده کنند. امید است مطالب این کتاب برای شما مفید باشد. خواهشمند است نظرات و پیشنهادات خود در خصوص این کتاب را از طریق ایمیل admin@Beyamooz با ما در میان بگذارید. © کلیه حقوق این کتاب محفوظ است. از انتشار این کتاب جداً خودداری کرده و برای تهیه آن فقط از طریق سایت **Beyamooz** اقدام فرمایید. همچنین توجه فرمایید که هزینه کتاب ها، **فقط برای استفاده یک نفر** در نظر گرفته شده است.

توجه: پیش نمایش زیر، تنها با هدف جلب اطمینان شما از محتوای اصلی تهیه شده است. بنابراین اگر مورد تأیید شما قرار گرفت لطفاً از طریق سایت Beyamooz.com اقدام به خرید فایل کامل آن نمایید.

لینک خرید کتاب های آموزش SAP

۱. [کتاب آموزش برنامه نویسی و توسعه SAP | آموزش SAP ABAP \(کتاب پیش رو\)](#)
۲. [کتاب آموزش نرم افزار مدیریت کیفیت SAP | ماژول QM در SAP](#)
۳. [کتاب آموزش پیاده سازی و پشتیبانی SAP | آموزش SAP BASIS](#)
۴. [کتاب آموزش مدیریت پایگاه داده SAP | آموزش SAP HANA](#)
۵. [کتاب آموزش نرم افزار منابع انسانی SAP | ماژول HR در SAP](#)
۶. [کتاب آموزش نرم افزار فروشگاه های SAP | ماژول SD در SAP](#)
۷. [کتاب آموزش حسابداری مدیریت SAP | ماژول CO در SAP](#)
۸. [کتاب آموزش نرم افزار حسابداری SAP | ماژول FI در SAP](#)
۹. [کتاب آموزش نگهداری و تعمیرات SAP | ماژول PM در SAP](#)
۱۰. [کتاب آموزش برنامه ریزی تولید SAP | ماژول PP در SAP](#)
۱۱. [کتاب آموزش انبارداری SAP | ماژول MM در SAP](#)
۱۲. [لیست همه کتاب های SAP](#)

موفق و پاینده باشید

فهرست مطالب

فصل ۱ : مقدمه برنامه نویسی ABAP ۱۱

| | |
|---|--------------|
| ABAP چیست؟ | ۱۱ |
| SAP S/4 HANA چیست؟ | ۱۱ |
| تاریخچه نسخه های ABAP | ۱۲ |
| ABAP Runtime Environment چیست؟ | ۱۳ |
| محل ذخیره Program های ABAP | ۱۳ |
| انواع Program های ABAP بر اساس نحوه اجرا | ۱۳ |
| محیط کدنویسی ABAP | ۱۴ |
| آشنایی با معماری SAP | در فایل اصلی |
| لایه های مختلف SAP | در فایل اصلی |
| آشنایی با اجزای معماری SAP S/4 HANA | در فایل اصلی |
| ساختار Program های ABAP | در فایل اصلی |
| قوانین نامگذاری Program | در فایل اصلی |
| الزامات ایجاد یک Program | در فایل اصلی |
| انواع Program های ABAP بر اساس نحوه کدگذاری | در فایل اصلی |
| دستورات در ABAP | در فایل اصلی |
| انواع دستورات ABAP | در فایل اصلی |
| Keyword چیست؟ | در فایل اصلی |
| انواع Keyword ها در ABAP | در فایل اصلی |
| نحوه نوشتن دستورات ABAP | در فایل اصلی |
| Comment ها در ABAP | در فایل اصلی |

فصل ۲ : محیط برنامه نویسی ABAP ۱۵

| | |
|-----------------------------|--------------|
| SAP GUI چیست؟ | در فایل اصلی |
| Log On (ورود به سیستم) | در فایل اصلی |
| آشنایی با محیط SAP GUI | در فایل اصلی |
| تغییر کلمه عبور | در فایل اصلی |
| ABAP Editor (ویرایشگر ABAP) | در فایل اصلی |

Log Off (خروج از سیستم) در فایل اصلی

فصل ۳: اولین برنامه ABAP ۱۶

ساختار دستورات ABAP ۱۶

ایجاد اولین برنامه ABAP ۱۶

اضافه کردن دستورات و ذخیره برنامه ۲۱

بررسی و رفع Syntax Error ها ۲۲

Active کردن برنامه ۲۴

آموزش اجرای برنامه ۲۵

فصل ۴: Data Type ها در ABAP در فایل اصلی

Data Type چیست؟ در فایل اصلی

دسته بندی Data Type ها در ABAP در فایل اصلی

Elementary Data Types (انواع داده ابتدایی) در فایل اصلی

Complex Types (نوع های پیچیده) در فایل اصلی

Structure Data Type (نوع داده ساختاری) در فایل اصلی

Internal Table Data Type (نوع داده جدول داخلی) در فایل اصلی

فصل ۵: Data Object ها در ABAP در فایل اصلی

Literal چیست؟ در فایل اصلی

Constant چیست؟ در فایل اصلی

انواع Constant در فایل اصلی

Elementary Constants (ثابت های ابتدایی) در فایل اصلی

Complex Constants (ثابت های مختلط) در فایل اصلی

Reference Constants (ثابت های مرجع) در فایل اصلی

متغیرها در ABAP در فایل اصلی

انواع متغیرها در ABAP در فایل اصلی

فصل ۶: متغیرها در ABAP در فایل اصلی

Local Variables (متغیرهای محلی) در ABAP در فایل اصلی

Static Variables (متغیرهای استاتیک) در ABAP در فایل اصلی

Reference Variables (متغیرهای ارجاع) در ABAP در فایل اصلی

System Variables (متغیرهای سیستمی) در ABAP در فایل اصلی

Structured Variables (متغیرهای ساختاری) در ABAP در فایل اصلی

فصل ۷: دستورات مقدماتی ABAP ۲۷

دستور SKIP – درج خط خالی در خروجی ۲۸

دستور ULINE – درج خط پیوسته در خروجی ۳۰

دستور NO-ZERO – حذف صفرها از سمت چپ اعداد ۳۱

دستور CONDENCE – حذف فضاهای خالی از رشته ۳۲

قالب بندی متون در ABAP در فایل اصلی

دستور Write در ABAP در فایل اصلی

رشته ها در ABAP در فایل اصلی

زیر رشته ها در ABAP در فایل اصلی

توابع کار با رشته ها در ABAP در فایل اصلی

عملیات روی رشته ها در ABAP در فایل اصلی

تاریخ و زمان در ABAP در فایل اصلی

Timestamp در ABAP در فایل اصلی

مدیریت پیام ها در ABAP در فایل اصلی

نمایش پیام ها با کد نویسی مستقیم در برنامه در فایل اصلی

نمایش پیام ها با استفاده از Message Class در فایل اصلی

انواع Operator ها در ABAP در فایل اصلی

Arithmetic Operator ها در ABAP در فایل اصلی

Bit Operator ها در ABAP در فایل اصلی

String Operator ها در ABAP در فایل اصلی

Relational Operator ها در ABAP در فایل اصلی

Boolean Operator ها در ABAP در فایل اصلی

فصل ۸: دستورات شرطی در ABAP در فایل اصلی

دستورات شرطی در ABAP در فایل اصلی

دستور IF در ABAP در فایل اصلی

دستور IF ELSE در ABAP در فایل اصلی

دستور IF تودرتو در ABAP در فایل اصلی

دستور CASE در ABAP در فایل اصلی

فصل ۹: دستورات حلقه در ABAP در فایل اصلی

حلقه ها در ABAP در فایل اصلی

حلقه While در ABAP در فایل اصلی

حلقه Do در ABAP در فایل اصلی

حلقه تودرتو در ABAP در فایل اصلی

فصل ۱۰: دستورات کنترل حلقه در ABAP در فایل اصلی

دستورات کنترل حلقه در ABAP در فایل اصلی

دستور CONTINUE در ABAP در فایل اصلی

دستور CHECK در ABAP در فایل اصلی

دستور EXIT در ABAP در فایل اصلی

فصل ۱۱: ABAP Data Dictionary در فایل اصلی

ABAP Data Dictionary چیست؟ در فایل اصلی

انواع اشیاء در ABAP Data Dictionary در فایل اصلی

Domain چیست؟ در فایل اصلی

آموزش ویرایش Domain در فایل اصلی

آموزش حذف Domain در فایل اصلی

Data Element چیست؟ در فایل اصلی

آموزش ایجاد Data Element در فایل اصلی

آموزش ویرایش Data Element در فایل اصلی

آموزش حذف Data Element در فایل اصلی

Structure چیست؟ در فایل اصلی

آموزش ایجاد Structure در فایل اصلی

آموزش ویرایش Structure در فایل اصلی

آموزش حذف Structure در فایل اصلی

جدول چیست؟ در فایل اصلی

آموزش ایجاد جدول در فایل اصلی

| | |
|-------------------------------|--------------|
| آموزش ویرایش جدول..... | در فایل اصلی |
| آموزش حذف جدول..... | در فایل اصلی |
| View چیست؟..... | در فایل اصلی |
| آموزش ایجاد View..... | در فایل اصلی |
| آموزش ویرایش View..... | در فایل اصلی |
| آموزش حذف View..... | در فایل اصلی |
| Search Help چیست؟..... | در فایل اصلی |
| آموزش ایجاد Search Help..... | در فایل اصلی |
| آموزش ویرایش Search Help..... | در فایل اصلی |
| آموزش حذف Search Help..... | در فایل اصلی |
| Lock Objects چیست؟..... | در فایل اصلی |
| آموزش ایجاد Lock Object..... | در فایل اصلی |
| آموزش ویرایش Lock Object..... | در فایل اصلی |
| آموزش حذف Lock Object..... | در فایل اصلی |

فصل ۲ : Internal Table ها در ABAP..... ۳۵

| | |
|---|--------------|
| Internal Table چیست؟..... | ۳۵ |
| Header Line چیست؟..... | ۳۶ |
| آموزش ایجاد Internal Table..... | ۳۷ |
| آموزش درج داده در Internal Table..... | ۴۱ |
| دستور INSERT – درج یک ردیف در Internal Table..... | ۴۲ |
| دستور INSERT LINES – درج چند ردیف در Internal Table..... | در فایل اصلی |
| دستور APPEND – درج داده در Internal Table..... | در فایل اصلی |
| دستور COLLECT – درج داده در Internal Table..... | در فایل اصلی |
| خواندن و پردازش اطلاعات Internal Table..... | در فایل اصلی |
| دستور READ TABLE..... | در فایل اصلی |
| دستور LOOP..... | در فایل اصلی |
| دستور MOVE – کپی فیلد به فیلد Internal Table..... | در فایل اصلی |
| دستور MOVE-CORRESPONDING – کپی تمام فیلدها در یک مرحله..... | در فایل اصلی |
| دستور MOVE – کپی اطلاعات جدول بدون Header Line..... | در فایل اصلی |

| | |
|--------------|---|
| در فایل اصلی | Header Line دارای جدول اطلاعات جدول |
| در فایل اصلی | INDEX .. DELETE - حذف رکوردهای جدول با ایندکس |
| در فایل اصلی | DELETE .. Key - حذف رکوردهای جدول با کلید |
| در فایل اصلی | DELETE .. WHERE - حذف رکوردهای جدول با شرط دلخواه |

فصل ۱۳ : برنامه نویسی ماژولار در ABAP

| | |
|--------------|-----------------------------|
| ۴۴ | برنامه نویسی ماژولار چیست؟ |
| ۴۵ | Macro ها در ABAP |
| ۴۹ | Include ها در ABAP |
| در فایل اصلی | Subroutine ها در ABAP |
| در فایل اصلی | Function Module چیست؟ |
| در فایل اصلی | مراحل ایجاد Function Module |
| در فایل اصلی | آموزش ایجاد Function Group |
| در فایل اصلی | آموزش ایجاد Function Module |
| در فایل اصلی | آموزش اجرای Function Module |

فصل ۱۴ : Screen ها در ABAP

| | |
|--------------|---|
| در فایل اصلی | SAPScript چیست؟ |
| در فایل اصلی | اجزای تشکیل دهنده SAPScript |
| در فایل اصلی | ایجاد SAPScript |
| در فایل اصلی | گنجاندن SAPScript در برنامه SAP |
| در فایل اصلی | Smart Form چیست؟ |
| در فایل اصلی | معماری Smart Form |
| در فایل اصلی | آموزش ایجاد Smart Form |
| در فایل اصلی | روش های توسعه SAP |
| در فایل اصلی | Customer Exit چیست؟ |
| در فایل اصلی | سفارشی سازی تیکد VD01 با استفاده از Customer Exit |
| در فایل اصلی | آموزش ایجاد Customer Exit |
| در فایل اصلی | User Exit چیست؟ |
| در فایل اصلی | آموزش ایجاد User Exit |

- در فایل اصلی BADIs (Business Add-Ins) چیست؟
- در فایل اصلی Web Dynpro چیست؟
- معماری Web Dynpro

فصل ۱۵ : ساخت گزارش ها در ABAP در فایل اصلی

- در فایل اصلی Report Program چیست؟
- در فایل اصلی گزارش های تعاملی چیست؟
- در فایل اصلی گزارش های ALV چیست؟
- در فایل اصلی انواع گزارش های ALV در SAP
- در فایل اصلی ساخت گزارش های ALV در SAP
- در فایل اصلی مرحله ۱: ایجاد یک Program جدید برای گزارش ALV
- در فایل اصلی مرحله ۲: ایجاد یک Structure جدید برای گزارش ALV
- در فایل اصلی مرحله ۳: برنامه نویسی گزارش ALV
- در فایل اصلی مرحله ۴: اضافه کردن فیلتر به گزارش ALV

فصل ۱۶ : Dialog Program ها در ABAP در فایل اصلی

- در فایل اصلی Dialog Program چیست؟
- در فایل اصلی ایجاد Dialog Program
- در فایل اصلی افزودن Screen به Dialog Program
- در فایل اصلی ایجاد تیکد برای Dialog Program

فصل ۱۷ : دستورات OPEN SQL در ABAP در فایل اصلی

- در فایل اصلی OPEN SQL چیست؟
- در فایل اصلی دستورات SQL DDL و SQL DML چیست؟
- در فایل اصلی لیست دستورات OPEN SQL
- در فایل اصلی دستور CLEAR در OPEN SQL
- در فایل اصلی دستور SELECT در OPEN SQL
- در فایل اصلی دستور INSERT در OPEN SQL
- در فایل اصلی دستور UPDATE در OPEN SQL
- در فایل اصلی دستور MODIFY در OPEN SQL
- در فایل اصلی دستور DELETE در OPEN SQL

فصل ۱۸ : دستورات Native SQL در ABAP در فایل اصلی

Native SQL چیست؟ در فایل اصلی

نحوه استفاده از دستورات Native SQL در فایل اصلی

فصل ۱۹ : برنامه نویسی شی گرا در ABAP ۵۲

برنامه نویسی شی گرا چیست؟ ۵۲

Object (شیء) چیست؟ ۵۲

Class چیست؟ ۵۲

نحوه تعریف Class در ABAP ۵۳

اجزاء تشکیل دهنده Class ۵۴

Class Attributes در فایل اصلی

Class Methods در فایل اصلی

Class Constructors در فایل اصلی

Class Events در فایل اصلی

Object چیست؟ در فایل اصلی

ایجاد Object در ABAP در فایل اصلی

Interface چیست؟ در فایل اصلی

تعریف Interface در فایل اصلی

پیاده سازی Interface در فایل اصلی

Encapsulation چیست؟ در فایل اصلی

Inheritance چیست؟ در فایل اصلی

قانون Single Inheritance چیست؟ در فایل اصلی

دسترسی به اجزاء Class در فایل اصلی

انواع Inheritance در فایل اصلی

Polymorphism چیست؟ در فایل اصلی

Exception چیست؟ در فایل اصلی

Exception Handling چیست؟ در فایل اصلی

فصل ۱ : مقدمه برنامه نویسی ABAP

در این کتاب آموزشی کلیه موضوعات برنامه نویسی ABAP با جزئیات پوشش داده شده است و برای همه افراد با اطلاعات کم یا حتی بدون دانش SAP ABAP مناسب است. قانداً مخاطبانی که تجربه برنامه نویسی و کار با پایگاه داده را دارند، مفاهیم را سریعتر درک خواهند کرد. کاربران بدون تجربه برنامه نویسی، ممکن است نیاز داشته باشند که موضوع را بیش از یک بار مرور کنند تا مفهوم را به وضوح درک کنند.

ABAP چیست؟

ABAP سرنام واژه های آلمانی **Allgemeiner Berichts Aufbereitungs Prozessor** بمعنی "پردازشگر تهیه گزارش عمومی (Generic Report Preparation Processor)" است. واژه مذکور بعد از گذشت چند سال به انگلیسی تبدیل شد و هم اکنون با عنوان "Advanced Business Application Programming" بمعنی "زبان برنامه نویسی برنامه های تجاری پیشرفته" شناخته می شود.

ABAP جزو زبان های نسل چهارم (4GLs = Fourth-Generation Languages) محسوب می شود که در دهه ۱۹۸۰ تأسیس شده است. ABAP در ابتدا بعنوان یک زبان گزارش ساز برای پلت فرم SAP R/2 توسعه یافت که توسط سازمان های بزرگ برای توسعه برنامه های مربوط به انبار و حسابداری استفاده می شد.

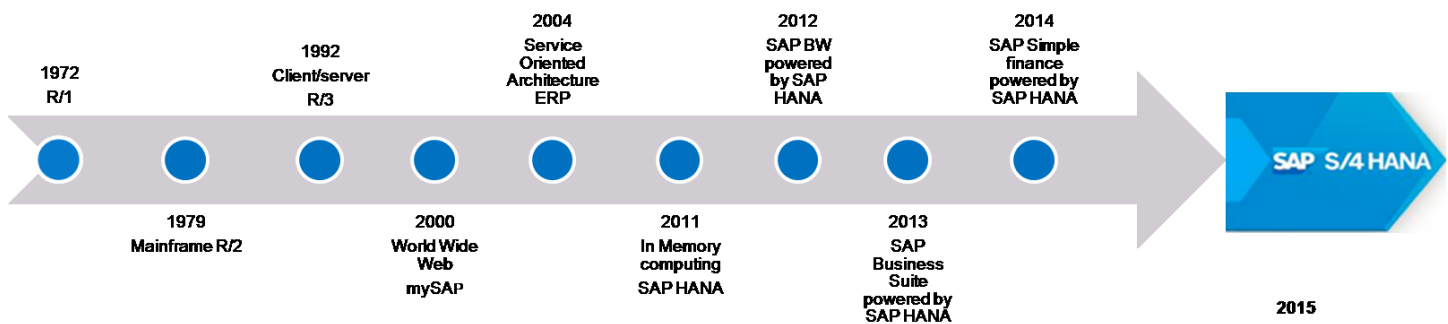
با روی کار آمدن نسخه SAP R/3، همچنان از زبان برنامه نویسی ABAP برای توسعه برنامه ها استفاده شد. خبر خوب اینکه در نسخه فعلی یعنی SAP S/4 HANA نیز از زبان برنامه نویسی ABAP و همچنین JAVA استفاده شده است.

چرا ABAP؟

- ABAP هم از روش برنامه نویسی رویه ای (Procedural) و هم شی گرا (Object-Oriented) پشتیبانی می کند.
- ABAP با هدف توسعه ی برنامه های SAP ایجاد شده است.
- ABAP به ما اجازه می دهد تا گزارش ها و فرم های مختلف را توسعه دهیم.
- ABAP از قابلیت حمل (Portability) پشتیبانی می کند و می تواند براحتی از یک پلت فرم به پلتفرم دیگر منتقل شود.
- ABAP برای افرادی که پیش زمینه برنامه نویسی دارند، بسیار آسان است.
- ABAP یکی از اولین زبان های برنامه نویسی است که مفهوم پایگاه های داده منطقی (LDBs = Logical Databases) را در بر دارد و از هر پلتفرم (Platform)، زبان (Language) و واحد (Units)، پشتیبانی می کند.
- با استفاده از ABAP می توان پلت فرم SAP S/4 HANA را توسعه داد.

SAP S/4 HANA چیست؟

SAP S/4 HANA آخرین نسخه SAP است. در زیر تاریخچه تکامل SAP S/4 HANA را مشاهده می فرمایید:



- اصطلاح HANA سرنام واژه های High Performance Analytic Appliance است.
- S/4 HANA چیست؟ آیا یک محصول کاملاً جدید SAP است؟ پاسخ خیر است. S/4 HANA مانند R/2 و R/3 است البته با خط کدهای جدید
- R/3 مخفف چیست؟
 - Real-Time System = R (سیستم بدون درنگ)
 - 3-Tier Architecture = 3 (معماری سه لایه)
- یکی از ویژگی های کلیدی SAP S/4 HANA طراحی in-memory technology platform است. این بدان معنی است که پایگاه داده کامل از قبل در حافظه اصلی است و لازم نیست قبل از تجزیه و تحلیل یا پردازش از یک هارد دیسک به حافظه اصلی کپی شود.
- آینده زبان برنامه نویسی ABAP چیست؟ خبر خوب برای همه ABAP کارها این است که S/4 HANA نیز از زبان برنامه نویسی ABAP استفاده می کند.
- معماری SAP S/4 HANA چیست؟ S/4 HANA اساساً بر اساس فناوری "HANA in-memory" توسعه یافته و مطابق با دستورالعمل های طراحی Fiori طراحی شده است.

تاریخچه نسخه های ABAP

- ABAP جزو زبان های نسل چهارم (4GLs = Fourth-Generation Languages) محسوب می شود که در دهه ۱۹۸۰ تأسیس شده است.
- در سال ۱۹۹۲ در نسخه SAP R/3 زبان برنامه نویسی ABAP بعنوان زبانی برای توسعه برنامه های "Client-Server" تبدیل می شود.
- در سال ۱۹۹۹، SAP افزونه "Object-Oriented" که با عنوان "ABAP Objects" نیز شناخته می شود را همراه با نسخه R/3 ۴,۶ منتشر کرد.
- در سال ۲۰۰۱، همه توابع SAP بجز ابتدایی ترین ها در ABAP نوشته شدند.
- در سال ۲۰۰۶، نسخه ۷,۰ منتشر شد که ویژگی "Switch Framework" را ارائه می دهد.

- در سال ۲۰۱۲، نسخه ۷,۴ با قابلیت "Table Expressions" منتشر شد.
- در سال ۲۰۱۵، نسخه ۷,۵ منتشر شد که شروع به پشتیبانی از "OPEN SQL Expressions" می کند.
- در سال ۲۰۱۷ نسخه ۷,۵۲ با قابلیت "Virtual Sorting" که از ویژگی "Internal Table" است منتشر شد.
- در سال ۲۰۱۹، نسخه ۷,۵۴ با بازنگری های زیادی در ویژگی های موجود منتشر شد.

ABAP Runtime Environment چیست؟

Runtime System (سیستم زمان اجرا) بخشی از هسته SAP است که اجرای برنامه های ABAP را کنترل می کند. Runtime System (سیستم زمان اجرا) مسئول پردازش دستورات ABAP، کنترل منطق جریان Screen ها (صفحه نمایش) و پاسخ به Event ها (رویدادها) است بعنوان مثال اگر کاربر روی یک دکمه روی صفحه کلیک کند، Runtime System (سیستم زمان اجرا) مسئول پردازش آن خواهد بود.

یکی از اجزای کلیدی Runtime System (سیستم زمان اجرا)، Database Interface (رابط پایگاه داده) است که دستورات ABAP مستقل از پایگاه داده ("OPEN SQL") را به عباراتی تبدیل می کند که توسط DBMS زیرین ("Native SQL") قابل درک است. رابط پایگاه داده تمام ارتباطات با پایگاه داده رابطه ای را از طرف برنامه های ABAP مدیریت می کند. رابط پایگاه داده همچنین دارای ویژگی هایی مانند بافر کردن جداول و داده هایی که مکرر به آنها رجوع می شود درحافظه محلی Application Server (سرور برنامه) است.

محل ذخیره Program های ABAP

همه Program های ABAP در پایگاه داده SAP ذخیره می شوند و مانند برنامه های JAVA یا C++ بعنوان یک فایل خارجی جداگانه ذخیره نمی شوند.

کد یک Program به دو شکل زیر در پایگاه داده ذخیره می شود:

۱- Source Code (سورس کد): این مدل کد را می توان با ابزارهای ABAP Workbench مشاهده و ویرایش کرد. (تیکد SE38)

۲- Generated Code (کد تولید شده): این مدل کد یک نمایش باینری است که کمی شبیه "Java Bytecode" است.

انواع Program های ABAP بر اساس نحوه اجرا

Program های ABAP از نظر اجرا (Execute) به دو نوع زیر تقسیم می شوند:

۱- Executable Programs (برنامه های قابل اجرا): این نوع Program ها بطور مستقل قابل اجرا هستند.

۲- Non Executable Programs (برنامه های غیر قابل اجرا): این نوع Program ها بطور مستقل قابل اجرا نیستند.

۱- Executable Programs (برنامه های قابل اجرا)

Executable Programs (برنامه های قابل اجرا) خود به دو نوع زیر تقسیم می شوند:

- **Reports (گزارش ها):** Program های از نوع Reports (گزارش ها) از یک مدل برنامه نویسی ساده پیروی می کنند که در آن کاربر مجموعه ای از پارامترهای ورودی را بصورت اختیاری وارد می کند و برنامه از آن پارامترهای ورودی برای تولید گزارش در قالب یک لیست تعاملی استفاده می کند. به دلیل ماهیت "لیست محور" خروجی برنامه، این برنامه ها با عنوان "Reports" شناخته می شوند.
- **Module pools (مخازن ماژول):** Program های از نوع Module pools (مخازن ماژول) راه های چالش برانگیز تعامل با کاربر را با استفاده از مجموعه ای از "Screen ها" (صفحه نمایش ها) ارائه می دهد. اصطلاح "Screen" تصویر واقعی که کاربر می تواند ببیند را مشخص می کند. هر "Screen" دارای یک "Flow Logic (منطق جریان)" است که بطور غیرمستقیم کد ABAP را فراخوانی می کند. این "Flow Logic (منطق جریان)" به دو بخش زیر تقسیم می شود:
 - ۱- "PBO (Process Before Output)": پردازش قبل از خروجی
 - ۲- "PAI (Process After Input)": پردازش پس از ورودیترکیب "Screen" و "Flow Logic (منطق جریان)" مربوط به آن برنامه ای با نام "Dynpro" یا Dynamic Program (برنامه پویا) بوجود خواهد آورد.

۲- Non Executable Programs (برنامه های غیر قابل اجرا)

Non Executable Programs (برنامه های غیر قابل اجرا) خود به انواع زیر تقسیم می شوند:

- **INCLUDE Modules (قرارگیری ماژول ها):** از این نوع Program بیشتر برای تقسیم بندی برنامه های بزرگ به واحدهای کوچکتر استفاده می شود. این واحدهای کوچک یعنی INCLUDE Module ها هنگام ایجاد برنامه های بزرگ در آنها صدا زده می شوند.
- **Subroutine Pools (مخزن زیر روال ها):** شامل Subroutine های ABAP است که بین FORM / ENDFORM کدگذاری شده و با دستور PERFORM می توان آنها را فراخوانی نمود.
- **Function Groups (گروه بندی توابع):** برای گروه بندی Function Module ها استفاده می شود. Function Module ها یا توابع را می توان با استفاده از دستور CALL FUNCTION فراخوانی نمود.
- **Object Classes (کلاس اشیاء):** مجموعه ای از Method ها و Attribute ها را تعریف می کند.
- **Interfaces (رابط ها):** شامل Definition (تعریف اولیه) Method های یک Class است.
- **Type Pools (مخزن نوع ها):** مجموعه ای از "Data Types (انواع داده ها)" و "Constants (ثابت ها)" را تعریف می کند.

توضیحات بیشتر در مورد انواع بالا در این کتاب بررسی خواهد شد.

محیط کدنویسی ABAP

با استفاده از یکی از دو محیط زیر می توان، برنامه های ABAP را توسعه داد:

۱- ABAP Workbench (میز کار ABAP)

۲- ABAP Development Tools (ابزارهای توسعه ABAP)

پایان پیش نمایش این فصل، ادامه در فایل اصلی ...!

توجه: پیش نمایش بالا، تنها با هدف جلب اطمینان شما از محتوای اصلی تهیه شده است. بنابراین اگر مورد تأیید شما قرار گرفت لطفاً از

طریق سایت Beyamooz.com اقدام به خرید فایل کامل آن نمایید.

لینک خرید کتاب های آموزش SAP

۱. [کتاب آموزش برنامه نویسی و توسعه SAP | آموزش SAP ABAP \(کتاب پیش رو\)](#)
۲. [کتاب آموزش نرم افزار مدیریت کیفیت SAP | ماژول QM در SAP](#)
۳. [کتاب آموزش پیاده سازی و پشتیبانی SAP | آموزش SAP BASIS](#)
۴. [کتاب آموزش مدیریت پایگاه داده SAP | آموزش SAP HANA](#)
۵. [کتاب آموزش نرم افزار منابع انسانی SAP | ماژول HR در SAP](#)
۶. [کتاب آموزش نرم افزار فروشگاههای SAP | ماژول SD در SAP](#)
۷. [کتاب آموزش حسابداری مدیریت SAP | ماژول CO در SAP](#)
۸. [کتاب آموزش نرم افزار حسابداری SAP | ماژول FI در SAP](#)
۹. [کتاب آموزش نگهداری و تعمیرات SAP | ماژول PM SAP](#)
۱۰. [کتاب آموزش برنامه ریزی تولید SAP | ماژول PP SAP](#)
۱۱. [کتاب آموزش انبارداری SAP | ماژول MM در SAP](#)
۱۲. [لیست همه کتاب های SAP](#)

موفق و پاینده باشید

فصل ۳: اولین برنامه ABAP

قبل از ادامه، اجازه دهید برخی از دستورالعمل های نوشتن یک Program (برنامه) را در ABAP بررسی کنیم.

ساختار دستورات ABAP

- هر عبارت در ABAP با یک کلمه کلیدی شروع می شود و با یک نقطه پایان می یابد.
- ABAP به حروف کوچک و بزرگ حساس نیست.
- اولین خط برنامه های گزارشی همیشه با کلمه کلیدی REPORT شروع می شوند. این نوع برنامه ها executable هستند.

```
1. REPORT [Program_Name].  
2. [Statements...].
```

- ABAP اهمیتی نمی دهد که یک عبارت از کجا شروع می شود. شما می توانید از این مزیت استفاده کنید و خوانایی برنامه خود را با استفاده از تورفتگی برای نشان دادن بلوک های کد، بهبود بخشید.
- ABAP هیچ محدودیتی در چیدمان عبارات ندارد. به این معنا که می توان چند دستور را روی یک خط قرار داد، یا ممکن است یک دستور در چندین خط کشیده شود.
- اگر ابتدای هر عبارت یکسان باشد، عبارات متوالی را می توان به هم متصل کرد. این کار با عملگر دونقطه (:) و کاما انجام می شود

```
1. WRITE 'Hello'.  
2. WRITE 'ABAP'.  
3. WRITE 'World'.  
4. -----  
5. WRITE: 'Hello',  
6.     'ABAP',  
7.     'World'.  
8. WRITE: 'Hello', 'ABAP', 'World'.
```

- برای Comment کردن تمام یک خط از کاراکتر * در ابتدای خط استفاده کنید. و برای Comment کردن قسمتی از یک خط از کاراکتر " استفاده کنید.

ایجاد اولین برنامه ABAP

همانطور که در مباحث قبلی گفته شد، SAP ABAP دارای یک ویرایشگر اختصاصی برای ایجاد برنامه های جدید است. برای شروع نوشتن برنامه جدید باید ABAP Editor را با استفاده از تیکد های SE38 یا SE80 باز کنیم.

در زیر مراحل کل فرآیند باز کردن ویرایشگر تا اجرای برنامه آمده است:

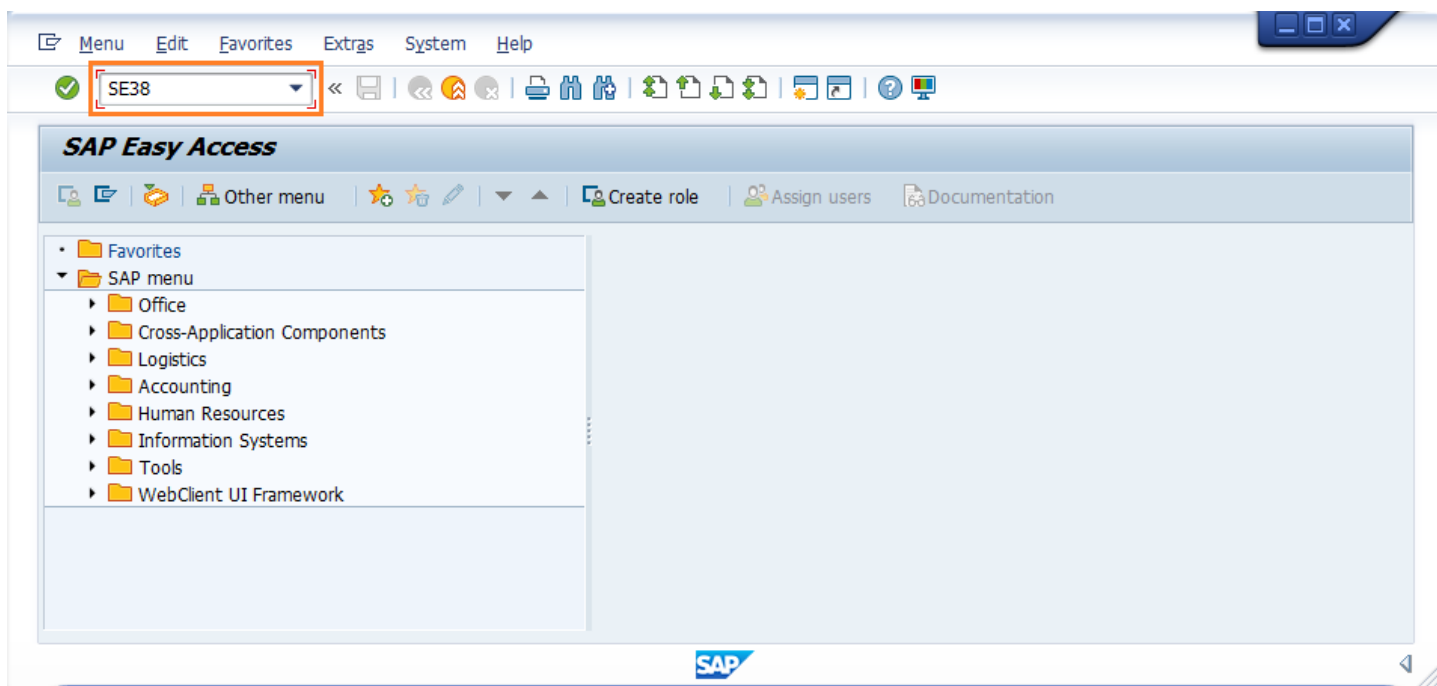
- باز کردن ABAP Editor
- اضافه کردن دستورات و ذخیره برنامه
- آموزش بررسی و رفع Syntax Error
- Active کردن برنامه
- اجرا کردن برنامه

اجازه دهید مراحل فوق را مرحله به مرحله مورد بررسی قرار دهیم.

باز کردن ABAP Editor

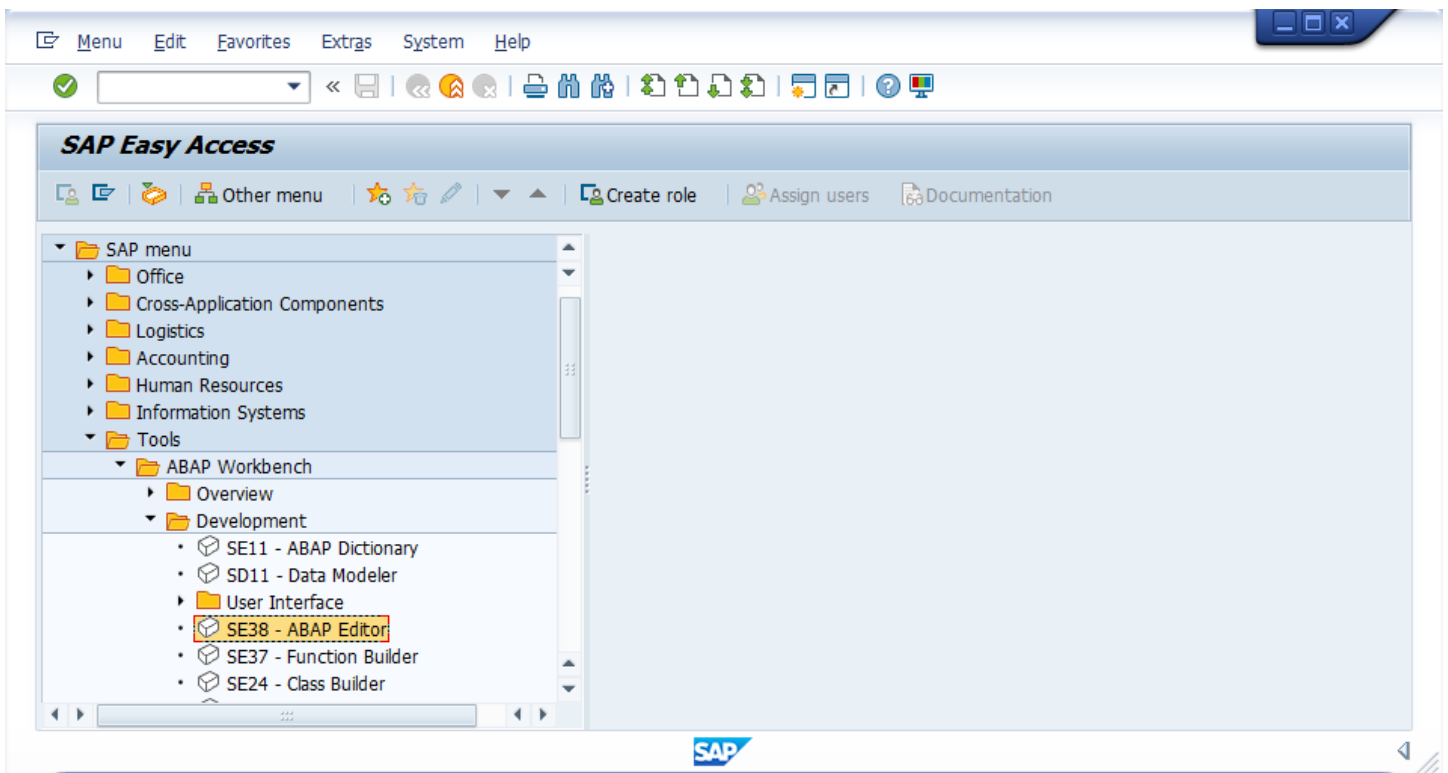
همان طور که در فصل قبل توضیح داده شد، با استفاده از یکی از دو روش زیر می توانیم ABAP Editor را باز کنیم:

۱- برای دسترسی به ABAP Editor، تیکد SE38 را در نوار فرمان وارد کرده و Enter کنید.



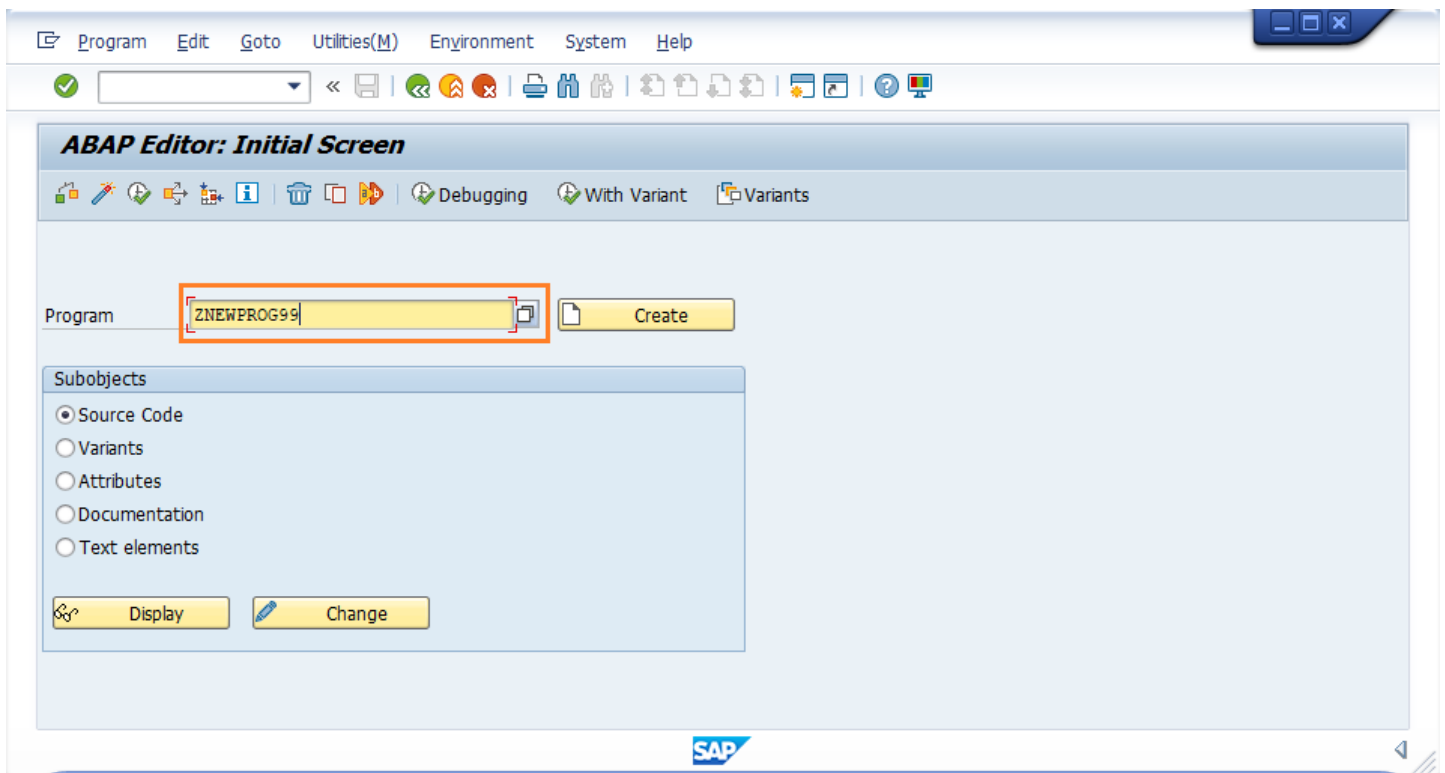
۲- برای دسترسی به ABAP Editor، مطابق تصویر زیر، از منوی اصلی SAP، مسیر زیر را پیمایش کنید:

SAP menu -> Tools -> ABAP Workbench -> Development SE38 -> ABAP Editor



بلافاصله صفحه "ABAP Editor: Initial Screen" نمایان خواهد شد. نام **Program** را وارد نمایید (ZNEWPROG99)، از قسمت SubObjects گزینه "Source Code" را انتخاب نمایید و روی دکمه "Create" کلیک نمایید تا به صفحه بعدی بروید.

نکته مهم! در ابتدای نام برنامه از کاراکتر "Z" استفاده شده است تا اطمینان حاصل شود که این Program سیستمی نیست و توسط مشتری ایجاد شده است. تمام اشیاء غیرسیستمی باید با پیشوند "Y" یا "Z" شروع شوند. "Y" یا "Z" باید همیشه پیشوند نام برنامه باشد تا شیء را با اشیاء سیستمی متمایز کند و از تداخل نام اشیاء جلوگیری کند.



بلافاصله پنجره "ABAP: Program Attributes ..." مانند زیر باز خواهد شد. نام برنامه بصورت اتوماتیک از مرحله قبلی در فیلد Title تنظیم می شود. اطلاعات مورد نیاز مانند Type (نوع برنامه) و ... را تنظیم کرده و برای ادامه روی دکمه Save کلیک نمایید.

ABAP: Program Attributes ZNEWPROG99 Change

Title **ZNEWPROG99**

Original language **EN** English

Created **06.05.2020** **PAWAN**

Last changed by

Status **New(Revised)**

Attributes

Type **Executable program**

Status

Application **Unknown application**

Authorization Group

Logical database

Selection screen

Editor lock Fixed point arithmetic

Unicode checks active Start using variant

Save

بلافاصله پنجره "Create Object Directory Entry" نمایان خواهد شد.

Create Object Directory Entry

Object **R3TR** **PROG** **ZNEWPROG99**

Attributes

Package

Person Responsible **PAWAN**

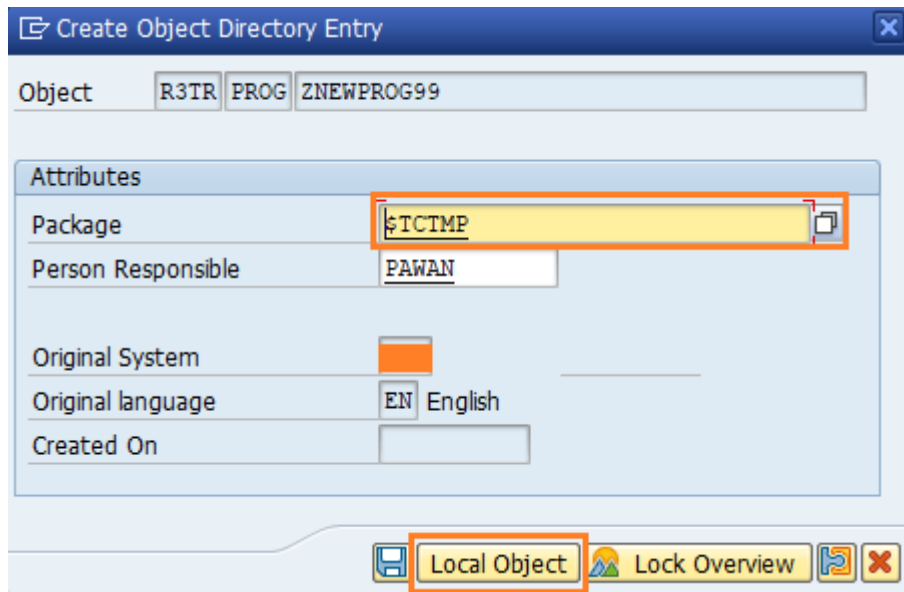
Original System

Original language **EN** English

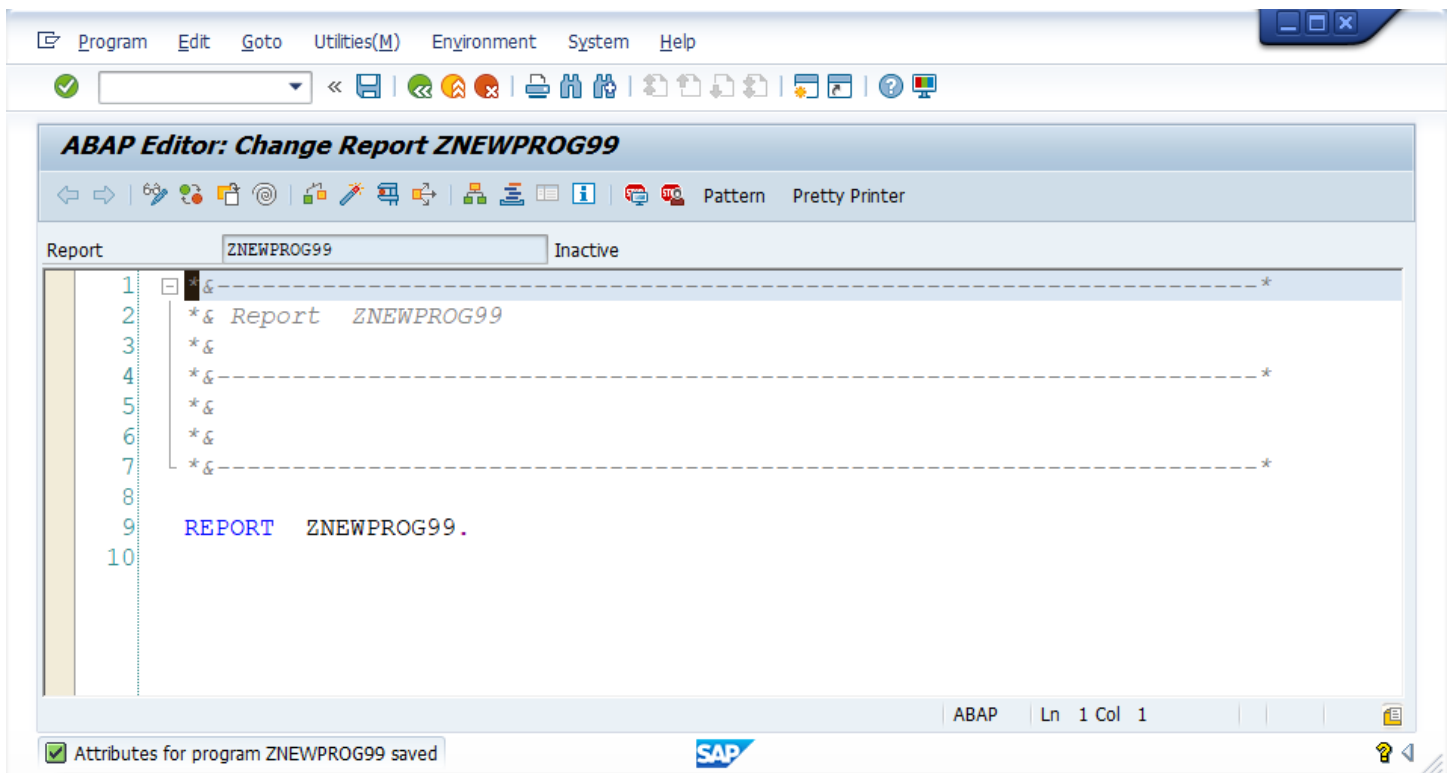
Created On

Local Object **Lock Overview**

پکیج (\$TCTMP) را انتخاب نمایید و روی دکمه "Local Object" یا "Save" کلیک نمایید تا ABAP Editor باز شود.

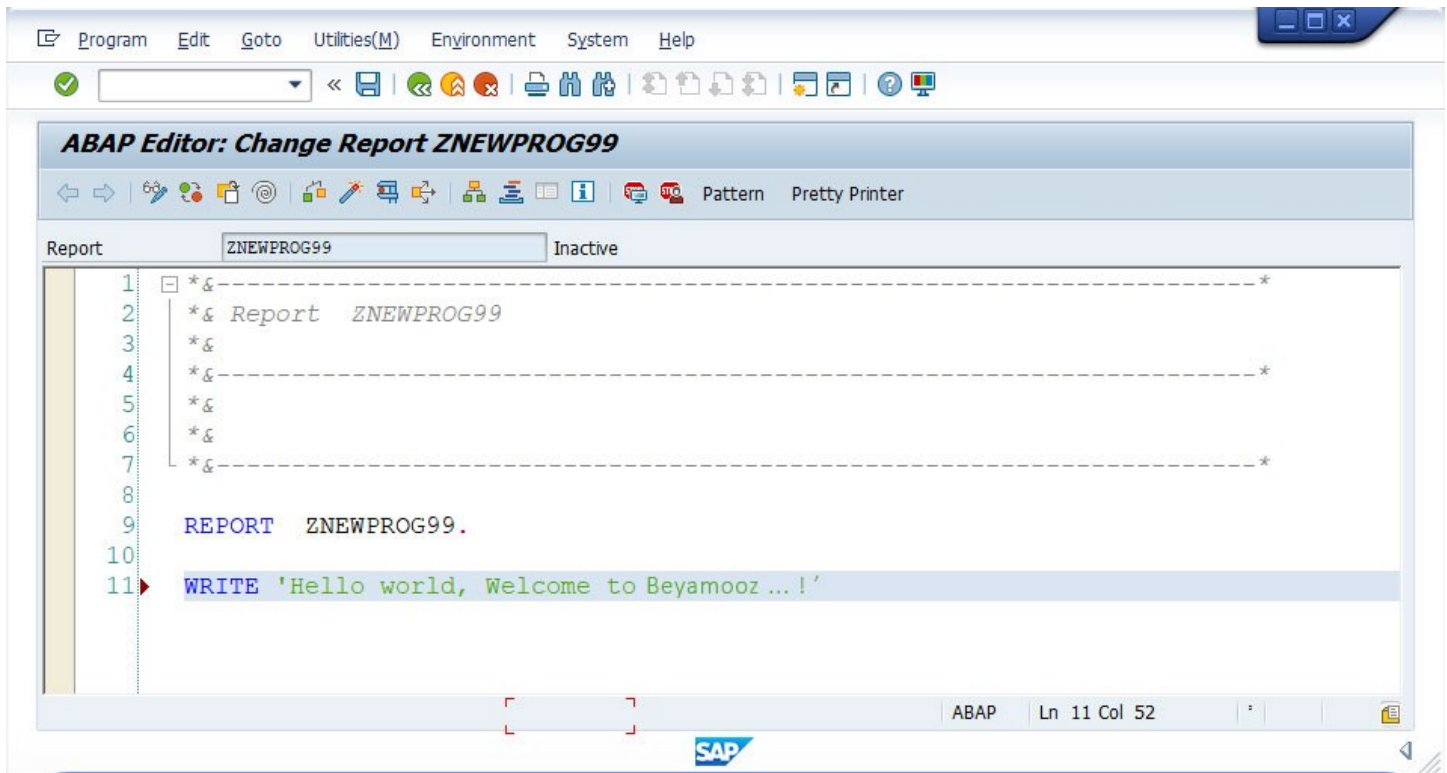


اکنون ABAP Editor برای برنامه ZNEWPROG99 مانند زیر باز شده است.

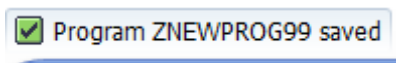


اضافه کردن دستورات و ذخیره برنامه

در اولین برنامه ABAP سعی داریم پیام "Hello world, Welcome to Beyamooz ... !" را در خروجی نمایش دهیم. بنابراین دستور Write را مطابق تصویر زیر به برنامه اضافه نمایید.



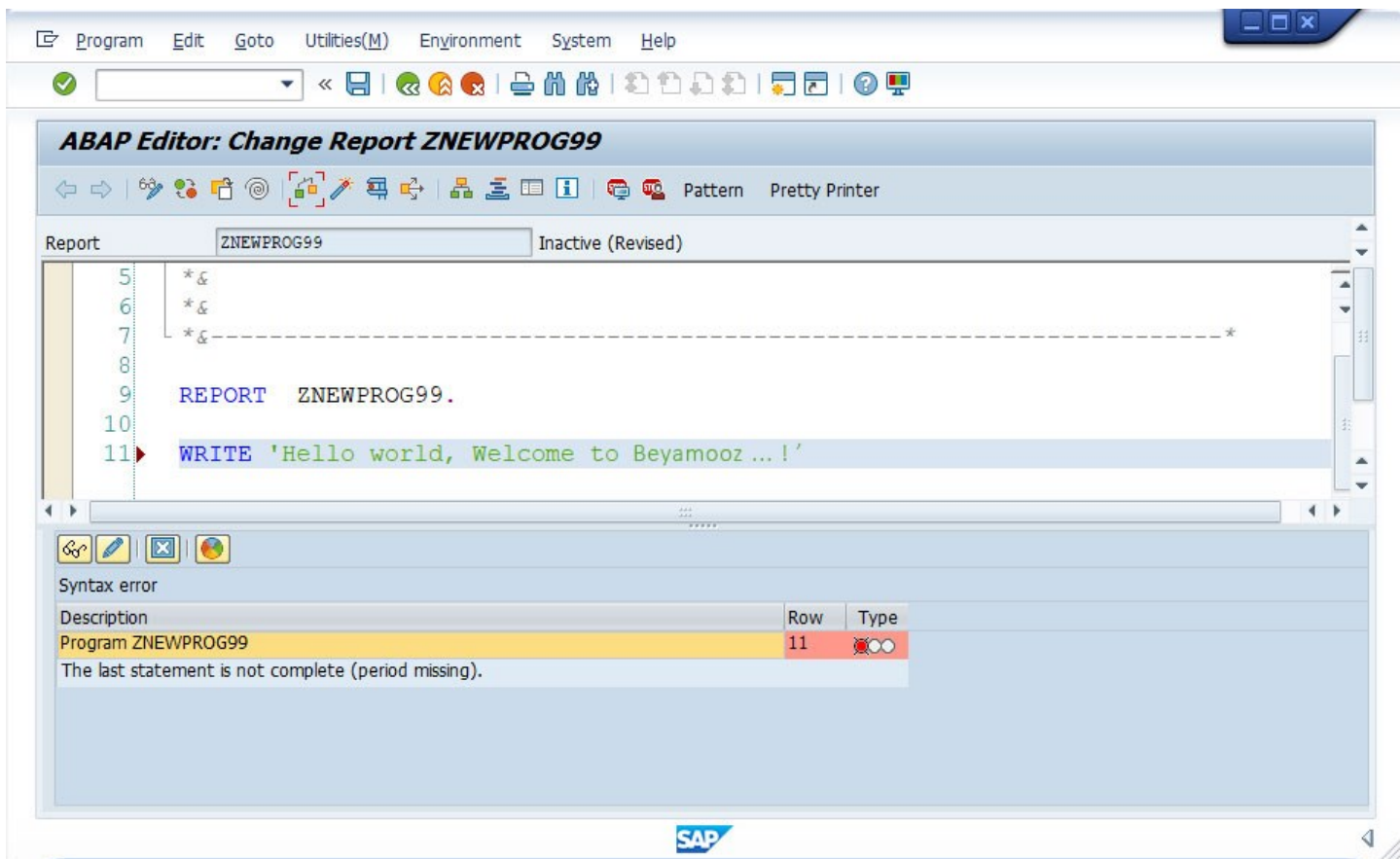
پس از اضافه کردن کد، روی دکمه **Save** در "نوار ابزار استاندارد" کلیک نمایید یا کلیدهای ترکیبی **"Ctrl + S"** را فشار دهید تا برنامه ذخیره شود. Status Bar را بررسی کنید و ادامه دهید.



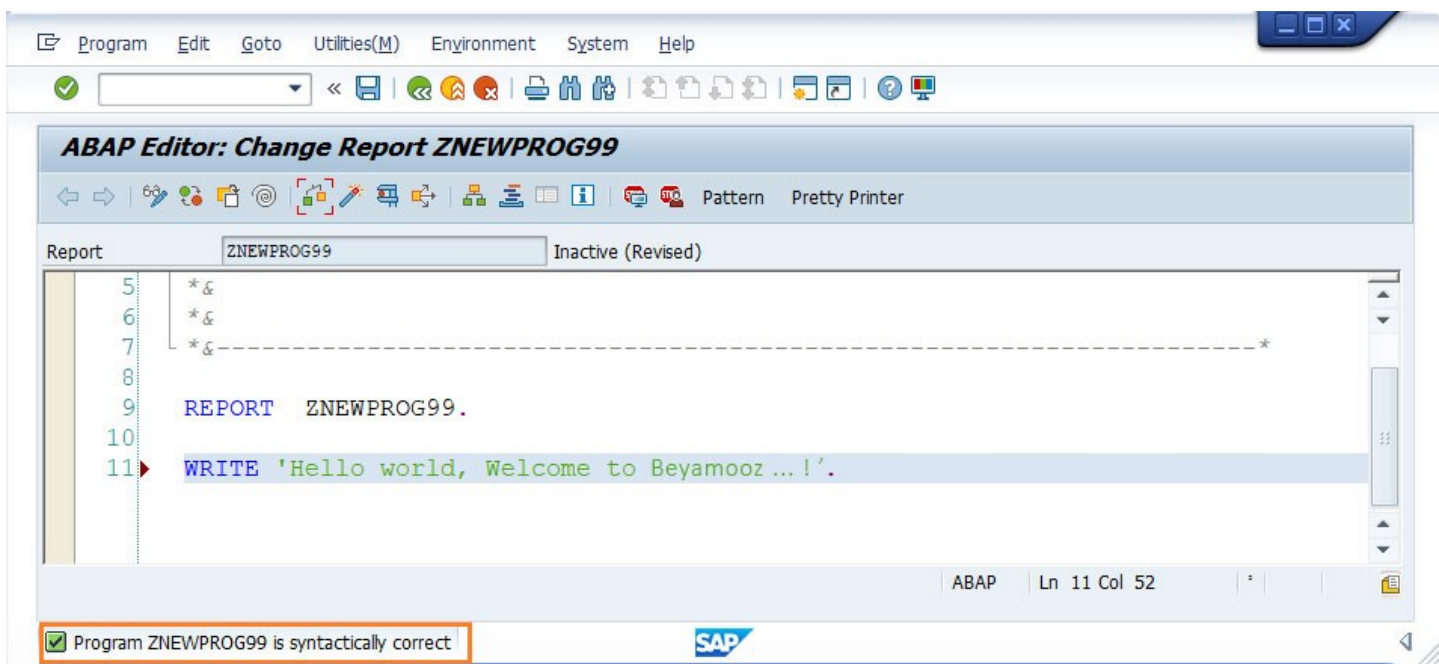
بررسی و رفع Syntax Error ها

برای بررسی Syntax Error (خطاهای نحوی) برنامه، کلیدهای ترکیبی **(Ctrl + F2)** را فشار دهید یا روی آیکن "Check" () در نوار ابزار برنامه کلیک نمایید.

در مثال بالا، ما عمداً نقطه (.) را در انتهای دستور Write نگذاشته ایم. بنابراین با خطاهای زیر مواجه خواهیم شد:

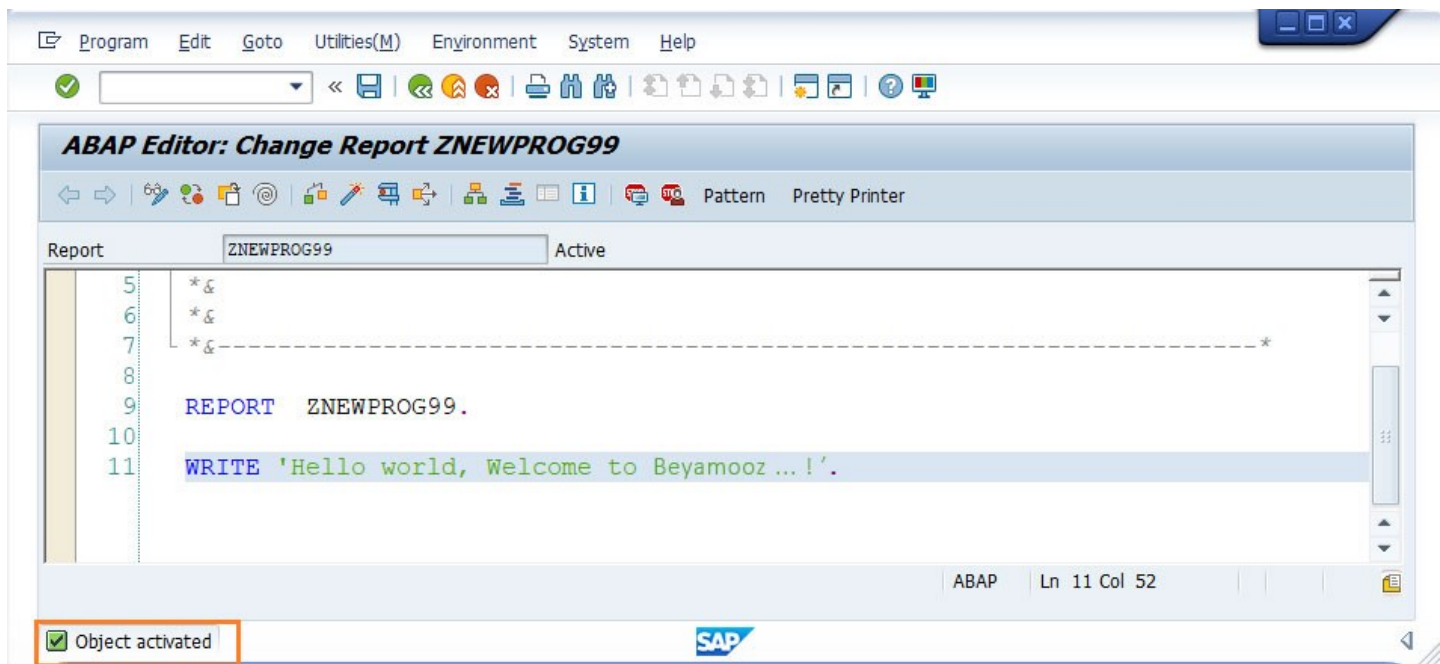


یک نقطه در انتهای دستور Write اضافه کنید و دوباره کلیدهای ترکیبی (Ctrl + F2) را فشار دهید. هنگامی که هیچ خطایی وجود نداشت، Status Bar مانند زیر نمایش داده خواهد شد.

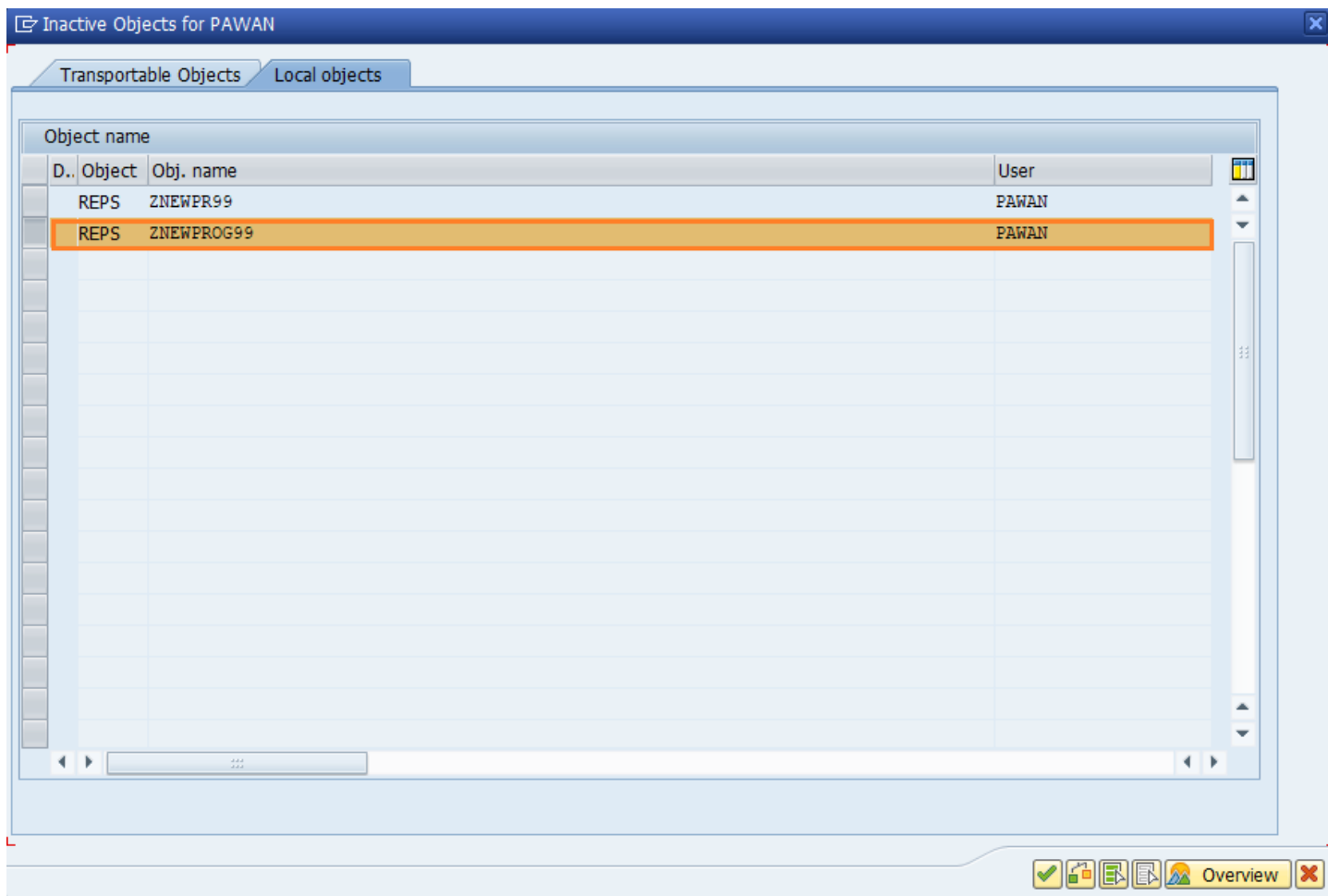


Active کردن برنامه

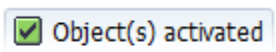
هنگامی که برنامه از نظر Syntax (نحوی) درست شد، با فشار دادن کلیدهای ترکیبی (Ctrl + F3) یا آیکن () "Activate" در نوار ابزار برنامه، برنامه را فعال کنید.




توجه: گاهی اوقات، در صورت وجود برنامه InActive (غیرفعال)، یک پنجره با تمام برنامه های InActive باز خواهد شد. از لیست، برنامه مورد نظر را انتخاب نمایید و روی آیکن "Continue" کلیک نمایید تا فعال شود.

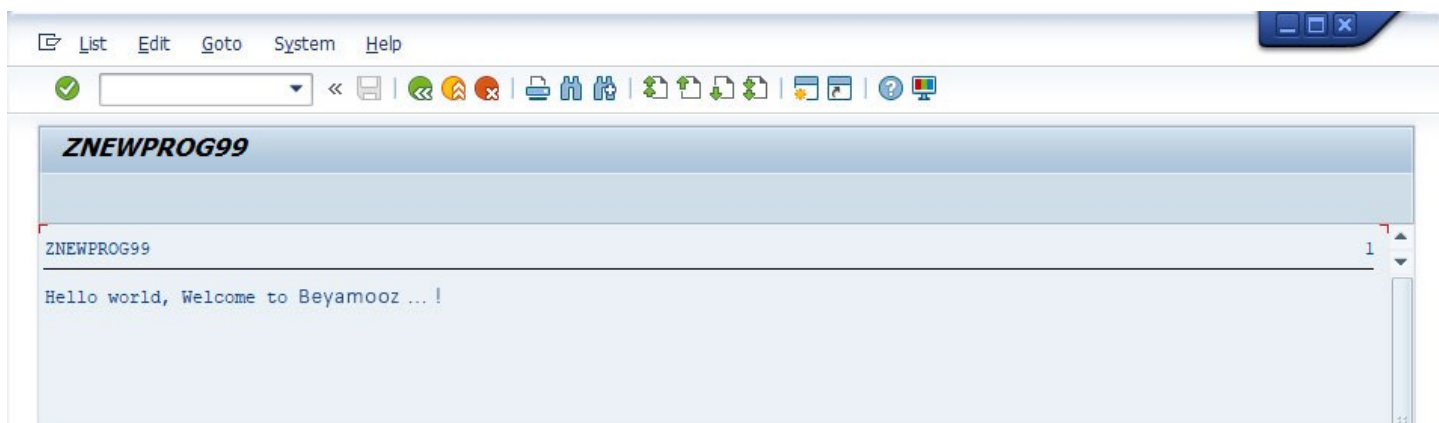


هنگامی که برنامه با موفقیت Active شود، در Status Bar پیام زیر نمایش داده خواهد شد:



آموزش اجرای برنامه

مرحله ۵: هنگامی که برنامه Active شد، برنامه را می توانید با فشار دادن "F8" یا کلیک روی آیکن "Direct Processing.." () اجرا نمایید. خروجی مانند زیر نمایش داده شود.



خروجی برنامه ZNEWPROG99 مانند بالا نمایش داده خواهد شد.

پایان پیش نمایش این فصل، ادامه در فایل اصلی ...!

توجه: پیش نمایش بالا، تنها با هدف جلب اطمینان شما از محتوای اصلی تهیه شده است. بنابراین اگر مورد تأیید شما قرار گرفت لطفاً از

طریق سایت Beyamooz.com اقدام به خرید فایل کامل آن نمایید.

لینک خرید کتاب های آموزش SAP

۱. [کتاب آموزش برنامه نویسی و توسعه SAP | آموزش SAP ABAP \(کتاب پیش رو\)](#)
۲. [کتاب آموزش نرم افزار مدیریت کیفیت SAP | ماژول QM در SAP](#)
۳. [کتاب آموزش پیاده سازی و پشتیبانی SAP | آموزش SAP BASIS](#)
۴. [کتاب آموزش مدیریت پایگاه داده SAP | آموزش SAP HANA](#)
۵. [کتاب آموزش نرم افزار منابع انسانی SAP | ماژول HR در SAP](#)
۶. [کتاب آموزش نرم افزار فروشگاهی SAP | ماژول SD در SAP](#)
۷. [کتاب آموزش حسابداری مدیریت SAP | ماژول CO در SAP](#)
۸. [کتاب آموزش نرم افزار حسابداری SAP | ماژول FI در SAP](#)
۹. [کتاب آموزش نگهداری و تعمیرات SAP | ماژول PM در SAP](#)
۱۰. [کتاب آموزش برنامه ریزی تولید SAP | ماژول PP در SAP](#)
۱۱. [کتاب آموزش انبارداری SAP | ماژول MM در SAP](#)
۱۲. [لیست همه کتاب های SAP](#)

موفق و پاینده باشید

فصل ۷: دستورات مقدماتی ABAP

برنامه های ABAP ترکیبی از دستورات Executable (قابل اجرا) و Non-Executable (غیرقابل اجرا) هستند. هر دستور Executable در ABAP با یک کلمه کلیدی، شروع می شود و با یک نقطه (.) به پایان می رسد. اولین دستور Executable با کلمه کلیدی REPORT و سپس نام گزارش (همان نام برنامه) شروع می شود و با نقطه (.) پایان می یابد. هنگام ایجاد برنامه، این دستور بطور خودکار توسط ویرایشگر تنظیم خواهد شد.

نحوه استفاده:

```
1. *&-----*
2. *& Report report_name
3. *&
4. *&-----*
5. *&
6. *&
7. *&-----*
8. REPORT report_name.
9. [Statement1].
10. [Statement2].
11. .
12. .
13. [Statement-n].
```

در دستور بالا، report_name (نام برنامه) قبلاً در هنگام ایجاد برنامه برای اولین بار توسط سیستم ABAP تنظیم خواهد شد.

بعنوان مثال:

فرض کنید ZNEWPROG99 برنامه جدید است. هنگام ایجاد برنامه ZNEWPROG99 با استفاده از ABAP Editor، دستور زیر بطور خودکار توسط ویرایشگر ایجاد می شود.

```
1. REPORT ZNEWPROG99.
```

تمام دستورات اجرایی را می توان بعد از دستور REPORT کدگذاری کرد. در زیر لیستی از دستورات مقدماتی آمده است که در ادامه مورد بررسی قرار خواهند گرفت:

- دستور SKIP - درج خط خالی در خروجی
- دستور ULINE - درج خط پیوسته در خروجی

- دستور NO-ZERO - حذف صفرها از سمت چپ اعداد
- دستور CONDENCE - حذف فضاهای خالی از رشته

دستور SKIP - درج خط خالی در خروجی

در زبان برنامه نویسی ABAP با استفاده از دستور SKIP می توان در خروجی بین خطوط یک Blank Line (خط خالی) درج نمود.

نحوه استفاده:

1. SKIP.
2. SKIP [no-of-lines].
3. SKIP [TO LINE line-number].

- SKIP: خط فعلی را رد می کند و خروجی بعدی از خط بعدی شروع به نمایش می کند.
- SKIP no-of-lines: به تعداد مشخص (no-of-lines) از خط فعلی رد می کند و خروجی پس از پرش از خط بعدی شروع به نمایش می کند.
- SKIP TO LINE line-number (پرش به شماره خط): خروجی را از شماره خط مشخص شده شروع کنید.

مثال:

نمایش پیام شماره خطوط برای خطوط ۳، ۵، ۱۰ و ۱۵.

1. *&-----*
2. *& Report ZSKIPLINE
3. *&-----*
4. *& Written by Beyamooz
5. *&-----*
6. REPORT ZSKIPLINE.
7. * First row in output occupies with program name
8. * Second row in output occupies with line
9. * What ever writing in the output, starts displaying from third line.
10. WRITE 'This is 3rd line'.
11. * Skip 4th line.
12. SKIP.
13. * Writing on the 5th row.
14. WRITE 'This is 5th line'.
15. * Skipping 4 lines.
16. SKIP 4.

17. * Wrting on the 10th row.
18. WRITE 'This is 10th line'.
19. * Moving the cursor to the 15th line.
20. SKIP TO LINE 15.
21. * Start writing on 15th row.
22. WRITE 'This is 15th line'.

خروجی:

```

ZSKIPLINE
This is 3rd line
This is 5th line

This is 10th line

This is 15th line

```

توضیح مثال:

در مثال بالا، قبل از هر دستور، یک Comment برای توضیح در مورد عبارت ارائه شده است. آنها را مرور کنید تا درک واضحی از کد نمونه بدست آورید.

- خط شماره ۱۰: ابتدا خروجی دستور WRITE در خط ۳ نمایش داده خواهد شد.
- خط شماره ۱۲: دستور SKIP از نمایش خط ۴ می گذرد.
- خط شماره ۱۴: دومین خروجی عبارت WRITE در خط ۵ نمایش داده خواهد شد. بنابراین در حال حاضر مکان نما در خط ۵ است.
- خط شماره ۱۶: SKIP 4 به تعدد 4 خط از خط ۵ را رد می کند و مکان نما را در خط ۱۰ قرار می دهد.
- خط شماره ۱۸: خروجی عبارت WRITE سوم در خط ۱۰ نمایش داده خواهد شد.
- خط شماره ۲۰: دستور SKIP TO LINE 15: مکان نما را در خط ۱۵ قرار می دهد.

- خط شماره ۲۲: چهارمین خروجی عبارت WRITE در خط ۱۵ نمایش داده خواهد شد.

توجه: همانطور که در فصل قبل گفته شد، هر خروجی برنامه ABAP از خط ۳ نمایش داده خواهد شد. خط اول با نام برنامه و خط دوم با خط پر شده است.

دستور ULINE – درج خط پیوسته در خروجی

در زبان برنامه نویسی ABAP با استفاده از دستور ULINE می توان یک خط پیوسته را در خروجی ایجاد نمود. ABAP از درج خط کامل و خط جزئی پشتیبانی می کند. درج خط جزئی نیاز به ورودی های بیشتری از کاربر دارد.

نحوه استفاده:

1. ULINE
2. ULINE AT starting_position(length)

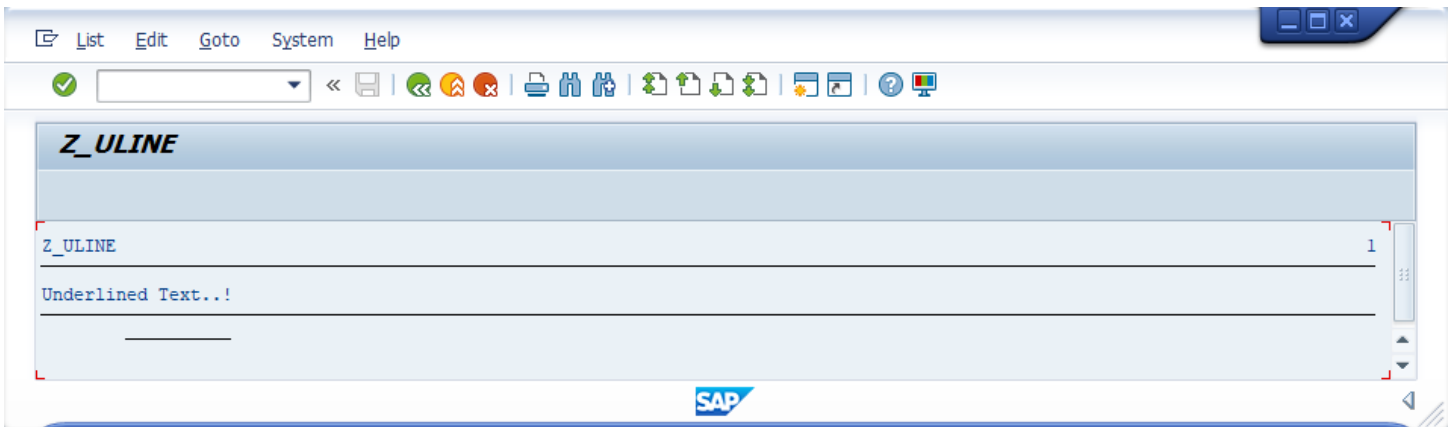
- ULINE: یک زیر خط پیوسته را در یک خط جدید درج می کند.
- ULINE AT starting_position(length): یک زیرخط از موقعیت شروع به طول مشخص شده درج می کند.

مثال:

برنامه ای بنویسید تا زیر خط کامل و جزئی را در خروجی نمایش دهد.

1. *&-----*
2. *& Report Z_ULINE
3. *&-----*
4. *& Written by Beyamooz
5. *&-----*
6. REPORT Z_ULINE.
7. WRITE 'Underlined Text..!'.
Underlined Text..!
8. ULINE.
9. ULINE AT 9(10).

خروجی:



توضیح مثال:

در مثال بالا، قبل از هر دستور، یک Comment برای توضیح در مورد عبارت ارائه شده است. آنها را مرور کنید تا درک واضحی از کد نمونه بدست آورید.

- خط شماره ۸: دستور ULINE خط کامل را بعد از خط دستور WRITE درج می کند.
- خط شماره ۱۰: دستور ULINE AT 9(10) از ستون ۹ به طول ۱۰ در خط فعلی درج می شود.

دستور NO-ZERO – حذف صفرها از سمت چپ اعداد

در زبان برنامه نویسی ABAP با استفاده از دستور NO-ZERO می توان صفرهای اضافی را از سمت چپ اعداد حذف نمود. به این ترتیب خوانایی و قابل فهم بودن را افزایش می دهیم. به این عمل Suppressing Zeroes (حذف صفرها) می گویند.

NO-ZERO فقط صفرها را در حین نمایش سرکوب یا حذف می کند. NO-ZERO صفرها را با چیزی جایگزین نمی کند. NO-ZERO فقط در فیلدهای عددی اعمال می شود.

نحوه استفاده:

1. WRITE numeric-field NO-ZERO.

مثال:

برنامه ای بنویسید تا مقدار عددی را با و بدون تابع NO-ZERO نمایش دهد.

1. *&-----*
2. *& Report Z_NOZERO
3. *&-----*

4. *& Written by Beyamooz
5. *&-----*
6. REPORT Z_NOZERO.
7. * Declaring W_N1 of type Numeric(N) with length 10
8. * and initializing with 70.
9. DATA: W_N1(10) TYPE N VALUE 70.
10. * Displaying the value without specifying NO-ZERO
11. WRITE : 'Displaying numeric data without NO-ZERO: ',W_N1.
12. * Skipping next two lines in the output.
13. SKIP 2.
14. * Displaying the value by specifying NO-ZERO
15. WRITE : 'Displaying numeric data with NO-ZERO: ',W_N1 NO-ZERO.

خروجی:

```

Z_NOZERO
-----
Z_NOZERO
-----
Displaying numeric data without NO-ZERO: 0000000070
-----
Displaying numeric data with NO-ZERO: 70

```

توضیح مثال:

در مثال بالا، قبل از هر دستور، یک Comment برای توضیح در مورد عبارت ارائه شده است. آنها را مرور کنید تا درک واضحی از کد نمونه بدست آورید.

- خط شماره ۹: متغیر W_N1 را با ۷۰ مقداردهی اولیه می کند.
- خط شماره ۱۱: نمایش مقدار بصورت کامل همراه با صفرهای ابتدایی.
- خط شماره ۱۵: نمایش مقدار بدون صفرهای ابتدایی.

دستور CONDENSE – حذف فضاهای خالی از رشته

از ABAP حذف فضای خالی (Space) از داده های ورودی پشتیبانی می کند. به این ترتیب خوانایی و قابل فهم بودن را افزایش می دهیم. دستور CONDENSE برای حذف فاصله ها از رشته ها استفاده می شود. به این عمل Suppressing Blanks (حذف فضاهای خالی) می گویند.

دستور CONDENSE فاصله های اضافی را از داده اصلی حذف خواهد کرد. CONDENSE فقط در فیلدهای کاراکتری اعمال می شود.

نحوه استفاده:

1. CONDENSE character-field [NO-GAPS].

• NO-GAPS: تمام فضاهای خالی را از رشته حذف می کند. اختیاری است.

مثال:

برنامه ای بنویسید تا رشته ای را با و بدون تابع CONDENSE نمایش دهد.

```
1. *&-----*
2. *& Report Z_CONDENSE
3. *&-----*
4. *& Written by Beyamooz
5. *&
6. *&-----*
7. REPORT Z_CONDENSE.
8. * Declaring variable of type character with length 50.
9. DATA: W_T1(50) TYPE C VALUE '  Tutorials Campus'.
10. * Displaying text before condense
11. WRITE : 'Text before condense: ',W_T1.
12. * Skipping next lines in the output
13. SKIP 2.
14. * Removing extra spaces from the string in w_t1.
15. CONDENSE W_T1.
16. * Display the extra spaces suppressed string.
17. WRITE : 'Text after condense: ',W_T1.
18. * Skipping next lines in the output
19. SKIP 2.
20. * Removing all spaces from the string in w_t1.
21. CONDENSE W_T1 NO-GAPS.
22. * Display all spaces removed string.
23. WRITE : 'Text after condense with NO_GAPS: ',W_T1.
```

```

Z_CONDENSE
-----
Text before condense -      Tutorials Campus

Text after condense -  Tutorials Campus

Text after condense with NO_GAPS - TutorialsCampus

```

پایان پیش نمایش این فصل، ادامه در فایل اصلی ...!

توجه: پیش نمایش بالا، تنها با هدف جلب اطمینان شما از محتوای اصلی تهیه شده است. بنابراین اگر مورد تأیید شما قرار گرفت لطفاً از طریق سایت Beyamooz.com اقدام به خرید فایل کامل آن نمایید.

لینک خرید کتاب های آموزش SAP

۱. [کتاب آموزش برنامه نویسی و توسعه SAP | آموزش SAP ABAP](#) (کتاب پیش رو)
۲. [کتاب آموزش نرم افزار مدیریت کیفیت SAP | ماژول QM در SAP](#)
۳. [کتاب آموزش پیاده سازی و پشتیبانی SAP | آموزش SAP BASIS](#)
۴. [کتاب آموزش مدیریت پایگاه داده SAP | آموزش SAP HANA](#)
۵. [کتاب آموزش نرم افزار منابع انسانی SAP | ماژول HR در SAP](#)
۶. [کتاب آموزش نرم افزار فروشگاه SAP | ماژول SD در SAP](#)
۷. [کتاب آموزش حسابداری مدیریت SAP | ماژول CO در SAP](#)
۸. [کتاب آموزش نرم افزار حسابداری SAP | ماژول FI در SAP](#)
۹. [کتاب آموزش نگهداری و تعمیرات SAP | ماژول PM در SAP](#)
۱۰. [کتاب آموزش برنامه ریزی تولید SAP | ماژول PP در SAP](#)
۱۱. [کتاب آموزش انبارداری SAP | ماژول MM در SAP](#)
۱۲. [لیست همه کتاب های SAP](#)

موفق و پاینده باشید

فصل ۱۲ : Internal Table ها در ABAP

Internal Table چیست؟

Internal Table (جدول داخلی) یک جدول موقت است که در حین اجرای برنامه، ایجاد و استفاده می شود و قبل از پایان برنامه حذف خواهد شد. در زبان برنامه نویسی ABAP از Internal Table ها برای ذخیره موقت اطلاعات داینامیک استفاده می شود. اطلاعات مذکور دارای یک ساختار ثابت هستند و از حافظه اصلی (Main Memory) به Internal Table ها منتقل خواهند شد.

طول عمر Internal Table تا پایان اجرای برنامه است. Internal Table ها در سایر زبان های برنامه نویسی بیشتر با عنوان آرایه شناخته می شوند. آرایه ها می توانند ساده یا آرایه ای از Structure ها باشند. در ABAP، آرایه ها را Internal Tables می نامند. Internal Table ها پیچیده ترین اشیاء داده در محیط ABAP هستند.

یک کاربرد مهم برای Internal Table ها، ذخیره و قالب بندی داده ها از یک جدول پایگاه داده در یک برنامه است.

چگونه داده ها در Internal Table ها ذخیره می شوند؟

داده ها در Internal Table بصورت سطر و ستون ذخیره می شوند. سطرها در Internal Table با نام ردیف و ستون ها با عنوان فیلد نامیده می شوند.

تمام ردیف ها در Internal Table دارای یک کلید و ساختار یکسان هستند. پس از پایان برنامه، Internal Table و داده های مرتبط با آن حذف می شوند. یک ردیف در Internal Table در محلی به نام Work Area (ناحیه کاری) قرار می گیرد ناحیه مذکور، ساختاری شبیه یک ردیف Internal Table دارد و فقط می تواند شامل یک ردیف داده باشد. اما جدول حاوی چندین ردیف داده با ساختار یکسان است.

تعداد ردیف ها یا اندازه Internal Table ها ثابت نیست. اندازه Internal Table ممکن است از برنامه ای به برنامه دیگر بر اساس نیاز متفاوت باشد. اما این Internal Table ها در حافظه اصلی ذخیره می شوند. بنابراین پیشنهاد می شود اندازه این جداول را کوچک نگهدارید تا در مصرف حافظه اصلی، صرفه جوی شود.

Internal Table ها با چهار ویژگی زیر مشخص می شوند:

- **Table Type (نوع جدول):** نحوه دسترسی ABAP به ردیف های جدول را مشخص می کند.
- **Field Type (نوع فیلد):** با هر نوع داده ABAP می توان فیلدهای جدول را اعلان نمود.
- **Uniqueness of Key (یکتا بودن کلید):** مشخص می کند که کلید جدول، Unique (یکتا) است یا خیر
- **Key Components (اجزای کلید):** مشخص می کند که کلید جدول از ترکیب کدام فیلدها تشکیل شده است و در واقع معیارهایی را مشخص می کنند که ردیف های جدول بر اساس آن مشخص می شوند.

در زبان برنامه نویسی ABAP سه نوع Internal Table وجود دارد که عبارتند از:

- Standard Internal Tables (جدول استاندارد)
- Sorted Internal Tables (جدول مرتب شده)
- Hashed Internal Tables (جدول هش شده)

تفاوت بین Internal Table ها

| Standard Internal tables (جدول استاندارد) | Sorted Internal Tables (جدول مرتب شده) | Hashed Internal tables (جدول هش شده) |
|---|--|--|
| این نوع جدول، Internal Table های پیش فرض هستند. | این جدول نوع خاصی از Internal Table ها هستند که هنگام درج رکورد، داده ها بطور خودکار مرتب می شوند. | این نوع جدول با Logical Databases (پایگاه داده های منطقی) استفاده می شوند. |
| دارای Index Table (جدول اینکدس) است. | دارای Index Table (جدول اینکدس) است. | دارای Index Table (جدول اینکدس) نیست. |
| جدول دارای کلید غیر یکتا است | جدول دارای کلید یکتا است | کلید جدول یکتا است و ورودی های تکراری مجاز نیست. |
| از عملگر KEY یا INDEX برای خواندن یک ردیف استفاده می شود. | از عملگر KEY یا INDEX برای خواندن یک ردیف استفاده می شود. | از عملگر KEY برای خواندن یک ردیف استفاده می شود. عملگر INDEX مجاز نیست. |
| از نوع جستجوهای Linear Search (جستجوی خطی) یا Binary Search (جستجوی باینری) برای پیدا کردن یک رکورد استفاده می شود. | از نوع جستجوی Binary Search (جستجوی باینری) برای پیدا کردن یک رکورد استفاده می شود. | از الگوریتم Hashed برای پیدا کردن یک رکورد استفاده می شود. |
| داده ها بصورت پیش فرض مرتب نیستند باید از عملیات مرتب سازی برای مرتب سازی داده ها استفاده نمود. | داده ها بصورت پیش فرض مرتب شده اند. | داده ها بصورت پیش فرض مرتب شده اند. |
| ردیف ها را می توان INSERT یا APPEND کرد. | ردیف ها را می توان INSERT کرد. | در ABAP همراه با پروژة های BI استفاده می شود. |

Header Line چیست؟

Header Line یک پارامتر اختیاری در هنگام اعلان Internal Table است و نقش کلیدی ایفا می کند تا به سیستم اطلاع دهد که آیا Work Area (منطقه کاری) باید بطور خودکار ایجاد شود یا خیر.

Work Area (ناحیه کاری) ساختاری مشابه یک ردیف Internal Table دارد و از محتوای آن برای درج در جدول و یا خواندن اطلاعات یک ردیف Internal Table و ریختن اطلاعات ردیف در Work Area استفاده خواهد شد.

Work Area با عنوان Header Line نیز نامیده می شود.

Internal Table را می توان به دو روش ایجاد کرد که عبارتند از:

- Internal Table ها با Header Line
- Internal Table ها بدون Header Line

Internal Table ها با Header Line

- سیستم بطور خودکار Work Area را ایجاد می کند.
- در صورت انجام هر گونه عملیات خواندن در Internal Table، داده ها بطور خودکار در Work Area قرار می گیرند.
- در صورت هر گونه عملیات درج یا ویرایش، داده ها از Work Area بارگیری و در Internal Table منعکس می شوند.

Internal Table ها بدون Header Line

- هیچ Work Area مرتبط با جدول وجود ندارد.
- هنگام دسترسی به جداول، Work Area باید به صراحت مشخص شود.
- دسترسی مستقیم به جداول امکان پذیر نیست.

آموزش ایجاد Internal Table

Internal Table (جدول داخلی) یک جدول موقت است که در حین اجرای برنامه، ایجاد و استفاده می شود و قبل از پایان برنامه حذف خواهد شد. بنابراین، Internal Table باید در برنامه ای که از آن استفاده می کند ایجاد شود.

دستور DATA برای اعلان یک Internal Table استفاده می شود. اعلان جدول باید بین عبارات BEGIN OF و END OF تکمیل شود.

از عبارت OCCURS برای تعیین تعداد ردیف های پیشفرض جدول استفاده می شود. اگر عدد OCCURS با صفر تنظیم شود، مشخص می کند که جدول در ابتدا هیچ رکوردی ندارد.

محدودیت مشخصی برای تعداد ردیف های جدول وجود ندارد.

چگونه یک Internal Table ایجاد کنیم؟

Internal Table را می توان بروش های زیر ایجاد نمود:

۱. ایجاد جدول با استفاده از دستور TYPE

۲. ایجاد جدول با استفاده از Structure موجود
 ۳. ایجاد جدول با استفاده از جدول موجود
- اجازه دهید یک به یک به بررسی موارد بالا بپردازیم.

۱- ایجاد جدول با استفاده از دستور TYPE

این فرآیند دو مرحله دارد. ساختار جدول ایجاد شده با دستور TYPE مطابق شکل زیر است:

1. TYPE BEGIN OF <structure-name>
2. .
3. <fields-declaration>
4. .
5. END OF <structure-name>.

توضیح: کد بالا یک Structure با نام structure-name ایجاد می کند. در ادامه می توان با استفاده از نام Structure یک Internal Table ایجاد نمود.

1. DATA <table-name> TYPE <structure-name> OCCURS n [WITH HEADER LINE].

توضیح: با استفاده از دستور بالا یک جدول با نام table-name از نوع structure-name با n ردیف ایجاد خواهد شد. گزینه WITH HEADER LINE اختیاری است. اگر جدول با WITH HEADER LINE اعلان شده باشد، Internal Table با Header Line است. اگر WITH HEADER LINE نادیده گرفته شود، Internal Table بدون Header Line است.

مثال:

یک جدول دانش آموزی با استفاده از Structure ایجاد نمایید.

1. TYPE BEGIN OF sstudent.
2. no TYPE I,
3. name(20) TYPE C,
4. dob(10) TYPE C,
5. END OF sstudent.
6. DATA tstudent TYPE sstudent OCCURS 10 WITH HEADER LINE.

توضیح مثال:

- خط شماره ۱: sstudent ساختاری است که با سه فیلد no, name و dob اعلان شده است.
- خط شماره ۶: جدول tstudent از نوع sstudent با ۱۰ ردیف اعلان شده است. جدول با WITH HEADER LINE اعلان شده است. بنابراین جدول یک Internal Table با Header Line است.

۲- ایجاد جدول با استفاده از Structure موجود

کد زیر نحوه ایجاد جدول با Structure موجود را نشان می دهد:

1. DATA <table-name> TYPE <structure-name> OCCURS n [WITH HEADER LINE].

توضیح: با استفاده از دستور بالا یک جدول با نام table-name از نوع structure-name با n ردیف (OCCURS) ایجاد خواهد شد. گزینه WITH Header LINE اختیاری است. اگر جدول با WITH HEADER LINE اعلان شده باشد، Internal Table با Header Line است. اگر WITH HEADER LINE نادیده گرفته شود، Internal Table بدون Header Line است.

مثال:

یک جدول دانشجو با استفاده از Structure دانشجو که موجود است ایجاد نمایید.

1. DATA tstudent TYPE sstudent OCCURS 10.

توضیح مثال:

جدول tstudent از نوع sstudent با ۱۰ ردیف اعلان شده است.

۳- ایجاد جدول با استفاده از جدول موجود

در کد زیر نحوه ایجاد جدول با استفاده از یک جدول موجود توضیح داده شده است:

1. DATA <table-name> TYPE <table1> OCCURS n [WITH HEADER LINE].

توضیح: با استفاده از دستور بالا یک جدول با نام table-name از نوع table1 با n رخداد (OCCURS) ایجاد خواهد شد. گزینه WITH Header LINE اختیاری است. اگر جدول با WITH HEADER LINE اعلان شده باشد، Internal Table با Header Line است. اگر WITH HEADER LINE نادیده گرفته شود، Internal Table بدون Header Line است.

بعنوان مثال:

1. DATA tstudent TYPE sstd OCCURS 10 WITH HEADER LINE.

توضیح مثال:

جدول tstudent از نوع یک جدول دیگر با نام sstd با ۱۰ خط اعلان شده است. ساختار جدول sstd برای tstudent اعمال می شود.

مثال:

مثال ساده برای ایجاد Internal Table اطلاعات محصول با ساختار زیر.

Pid:Character (10)

Pname: Character (40)

Pamount: Decimal (10)

```
1. * &-----*
2. * & Report Z_INTERNAL_TABLE
3. * &-----*
4. * & Written by Beyamooz
5. * &-----*
6. REPORT Z_INTERNAL_TABLE.
7. * Internal table Structure creation
8. TYPES: BEGIN OF t_product,
9.     pid(10) TYPE C,
10.    pname(40) TYPE C,
11.    pamount(10) TYPE P,
12. END OF t_product.
13. * Data & internal table declaration
14. DATA: wa TYPE t_product,
15.       it TYPE TABLE OF t_product.
16. wa-pid = 'IFB1'.
17. wa-pname = 'IFB WASHING MACHINE'.
18. wa-pamount = 31000.
19. * Appending data to the internal table
20. APPEND wa TO it.
21. * Reading internal table of index 1
22. READ TABLE it INTO wa INDEX 1.
```


23. * IF sy-subrc equal to ZERO, the internal table read is successful.
 24. * Otherwise, 'No Record Found' gets displayed.
 25. IF sy-subrc = 0.
 26. WRITE: wa-pid, wa-pname, wa-pamount.
 27. ELSE.
 28. WRITE 'No Record Found'.
 29. ENDIF.

خروجی:

| Z_INTERNAL_TABLE | | |
|------------------|---------------------|--------|
| Z_INTERNAL_TABLE | | |
| IFB1 | IFB WASHING MACHINE | 31.000 |

توضیح مثال:

در مثال بالا، قبل از هر دستور، یک Comment برای توضیح در مورد عبارت ارائه شده است. آنها را مرور کنید تا درک واضحی از کد نمونه بدست آورید.

- خط شماره ۸: ساختار t_product همراه با سه فیلد اعلان شد.
- خط شماره ۱۴: متغیر wa از نوع ساختار t_product اعلان شد.
- خط شماره ۱۵: متغیر it که یک Internal Table است از نوع t_product اعلان شد.
- خط شماره ۲۰: با استفاده از دستور wa TO it محتوای Work Area (ناحیه کاری) (wa) به جدول (it) اضافه خواهد شد.
- خط شماره ۲۲: خواندن ایندکس شماره ۱ متغیر it که یک Internal Table است و قراردادن آن در wa

آموزش درج داده در Internal Table

Internal Table ها برای پردازش داده ها بصورت داینامیک در طول اجرای برنامه استفاده می شوند. بنابراین، درج داده ها در Internal Table را می توان در طول اجرای برنامه انجام داد. درج داده ها را می توان به سه روش زیر انجام داد:

۱- دستور INSERT

۲- دستور APPEND

دستور INSERT – درج یک ردیف در Internal Table

در زبان برنامه نویسی ABAP با استفاده از دستور INSERT می توان یک ردیف در Internal Table درج نمود. داده ها ابتدا باید در یک Work Area تنظیم شوند و دستور INSERT از همان Work Area برای درج داده ها در Internal Table استفاده خواهد کرد. از کلمه کلیدی INDEX برای تعیین موقعیت ردیفی که در آن سطر جدید درج می شود استفاده می شود.

اگر جدول قبلاً رکوردی در اینکدس مشخص شده داشته باشد، سیستم ردیف فعلی را در موقعیت مشخص شده درج می کند و ردیف های موجود را از موقعیت جاری، مجدداً با افزایش اینکدس آنها (+1) درج می کند.

هنگام استفاده از دستور INSERT اگر گزینه INDEX را با مقدار ۱- تنظیم نمایید، ردیف در انتهای جدول اضافه خواهد شد.

برای Standard Tables (جداول استاندارد)، درج ردیف ها با یک کلید غیر یکتا، بدون در نظر گرفتن تکراری بودن یا نبودن ردیف جاری، با موفقیت انجام خواهد شد که البته ردیف های تکراری در جدول بوجود خواهد آمد.

برای Sorted Tables (جداول مرتب شده)، درج ردیف ها با یک کلید غیر یکتا، بدون در نظر گرفتن تکراری بودن یا نبودن ردیف جاری، با موفقیت انجام خواهد شد که البته ردیف های تکراری در جدول بوجود خواهد آمد.

برای Sorted Tables (جداول مرتب شده)، درج ردیف ها با یک کلید یکتا، اگر کاربر تلاش کند ردیف را با کلید تکراری اضافه کند، خطای زمان اجرا ایجاد خواهد شد. همچنین اگر کاربر ترتیب مرتب سازی Sorted Table را نقض کند، خطای زمان اجرا را ایجاد می کند.

نحوه استفاده:

```
1. INSERT [<work-area>/INITIAL LINE] INTO TABLE <internal-table>.
```

یا

```
1. INSERT [<work-area>/INITIAL LINE] INTO <internal-table>
2. INDEX <index-num>.
```

- **<Work Area>**: Work Area (ناحیه کاری) ای را مشخص می کند که ساختاری مشابه یک ردیف Internal Table دارد و محتوای آن در جدول درج خواهد شد.
- دستور اول برای اضافه کردن Work Area در انتهای Internal Table استفاده می شود.

- دستور دوم برای اضافه کردن Work Area در اینکدس شماره index-num استفاده می شود. در واقع در دستور دوم باید موقعیتی را که می خواهیم ردیف در آنجا درج شود را تنظیم نماییم.
- اگر بجای Work Area از گزینه INITIAL LINE استفاده کنیم یک ردیف خالی همراه با مقادیری که هنگام تعریف جدول تنظیم کرده ایم در Internal Table درج خواهد داد.

پایان پیش نمایش این فصل، ادامه در فایل اصلی ...!

توجه: پیش نمایش بالا، تنها با هدف جلب اطمینان شما از محتوای اصلی تهیه شده است. بنابراین اگر مورد تأیید شما قرار گرفت لطفاً از

طریق سایت Beyamooz.com اقدام به خرید فایل کامل آن نمایید.

لینک خرید کتاب های آموزش SAP

۱. [کتاب آموزش برنامه نویسی و توسعه SAP | آموزش SAP ABAP \(کتاب پیش رو\)](#)
۲. [کتاب آموزش نرم افزار مدیریت کیفیت SAP | ماژول QM در SAP](#)
۳. [کتاب آموزش پیاده سازی و پشتیبانی SAP | آموزش SAP BASIS](#)
۴. [کتاب آموزش مدیریت پایگاه داده SAP | آموزش SAP HANA](#)
۵. [کتاب آموزش نرم افزار منابع انسانی SAP | ماژول HR در SAP](#)
۶. [کتاب آموزش نرم افزار فروشگاههای SAP | ماژول SD در SAP](#)
۷. [کتاب آموزش حسابداری مدیریت SAP | ماژول CO در SAP](#)
۸. [کتاب آموزش نرم افزار حسابداری SAP | ماژول FI در SAP](#)
۹. [کتاب آموزش نگهداری و تعمیرات SAP | ماژول PM در SAP](#)
۱۰. [کتاب آموزش برنامه ریزی تولید SAP | ماژول PP در SAP](#)
۱۱. [کتاب آموزش انبارداری SAP | ماژول MM در SAP](#)
۱۲. [لیست همه کتاب های SAP](#)

موفق و پاینده باشید

فصل ۱۳ : برنامه نویسی ماژولار در ABAP

برنامه نویسی ماژولار چیست؟

برنامه نویسی ماژولار، یک تکنیک برنامه نویسی است که در آن برنامه های بزرگ را به بخش های کوچکتر تقسیم می کنیم. هر کدام از این بخش های کوچک، وظایف و مسئولیت های خود را دارد.

در برنامه نویسی ماژولار، برنامه بصورت مجموعه ای از ماژول ها، سازماندهی می شود. این رویکرد امکان توسعه و تغییرات مستقل هر کد در برنامه را فراهم می کند، بدون اینکه نیاز به تغییر کلی برنامه وجود داشته باشد.

بعنوان مثال اگر چند متغیر یکسان در چندین برنامه استفاده می شود، می توان آنها را در واحد Modularization کدگذاری نمود و در برنامه های متعددی که مورد نیاز است از آن استفاده نمود.

مزایای برنامه نویسی ماژولار

- Readability (قابلیت خوانایی) و درک بهتر کد را افزایش می دهد.
- Maintain (نگهداری) کد را آسان تر می کند.
- Debug (اشکال زدایی) را آسان تر می کند.
- Redundancy (افزونگی) کد را حذف می کند.
- Reusability (قابلیت استفاده مجدد) کد را افزایش می دهد.
- Reliability (قابلیت اطمینان) را افزایش می دهد.

راه های پیاده سازی برنامه نویسی ماژولار در ABAP

در ABAP با استفاده از روش های زیر می توان برنامه نویسی ماژولار را پیاده سازی نمود:

| روش های برنامه نویسی ماژولار | توضیحات |
|------------------------------|--|
| Macros | اگر مجموعه ای از عبارات یکسان بیش از یک بار در برنامه تکرار شوند، می توان از آن عبارات برای ایجاد یک Macro استفاده نمود. Macro فقط در برنامه ای که در آن تعریف شده استفاده می شود. |
| Subroutines | Subroutine ها رویه هایی هستند که می توان آنها را در هر برنامه ای تعریف کرد و از هر برنامه ای فراخوانی نمود. بلوک دستورات Subroutine بین دستورات FORM و ENDFORM تعریف می شود. دستور PERFORM برای فراخوانی Subroutine استفاده می شود. هنگام فراخوانی Subroutine باید همان تعداد پارامتر که در FORM هنگام تعریف Subroutine در نظر گرفته ایم، ذکر شود. |

| | |
|------------------|---|
| Include programs | <p>اگر مجموعه ای از دستورات (سورس کد) در بیش از یک برنامه استفاده شود، آن مجموعه دستورات را می توان تحت قالب یک Include Program با یک نام یکتا، ذخیره نمود و سپس نام Include Program مذکور را در دیگر برنامه ها گنجانند.</p> <p>Include Program ها فقط برای شکستن برنامه های بزرگ به چند برنامه کوچکتر، استفاده می شوند و نمی توان برای آنها پارامتر در نظر گرفت. Include Program ها برنامه های مستقل نیستند و نمی توانند بطور مستقل اجرا شوند.</p> |
| Function modules | <p>Function Module ها، زیربرنامه های هستند که شامل مجموعه ای از دستورات (سورس کد) قابل استفاده مجدد هستند. برخلاف Include ها، Function Module ها دارای Import Parameter (پارامترهای ورودی) و Export Parameter (پارامترهای خروجی) هستند. همچنین می توان هنگام تعریف آنها، Exception (استثنا) را مدیریت نمود.</p> <p>Function Module ها برای کل سیستم در دسترس هستند.</p> |
| Methods/Classes | <p>با استفاده از Methods/Classes می توان برنامه نویسی شی گرا را به ABAP اضافه نمود.</p> <p>با استفاده از Methods/Classes می توان مفهوم Parameter Interface را پیاده سازی نمود.</p> |

Macro ها در ABAP

اگر یک بلوک دستورات را بیش از یک بار در مکان های مختلف برنامه تکرار کنیم، باعث افزونگی کد خواهیم شد. بلوک دستورات مذکور را می توان در Macro قرار داد و در جاهای مورد نیاز آنرا فراخوانی نمود. بنابراین Macro جایگزین بلوک دستورات شده و از تکرار مکرر کد جلوگیری خواهد کرد.

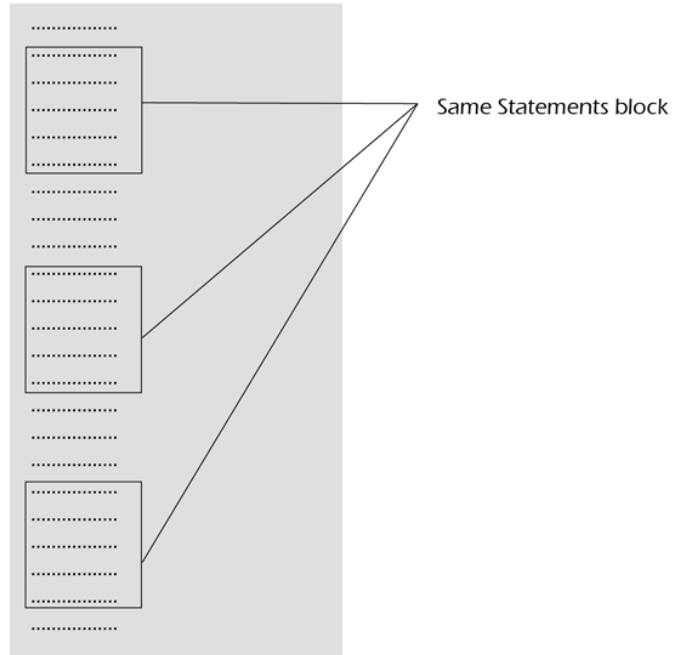
Macro یک بلوک دستور است که با مجموعه ای از دستورات ایجاد شده است و چندین بار در یک برنامه، آنرا فراخوانی می کنیم.

Macro فقط در برنامه ای که در آن تعریف شده استفاده می شود. مجموعه دستورات Macro باید قبل اینکه آنرا فراخوانی کنیم، تعریف شود. تمام عبارات موجود در Macro باید بین DEFINE و END-OF-DEFINITION کد گذاری شوند.

Macroها دستورهای عملیاتی نیستند. Macro نمی تواند خودش را صدا بزند. Macroها تودرتو هستند، Macro دیگری را می توان در یک MACRO فراخوانی کرد.

مثال زیر، برنامه Y را یکبار با Macro و یکبار بدون استفاده از Macro توضیح می دهد:

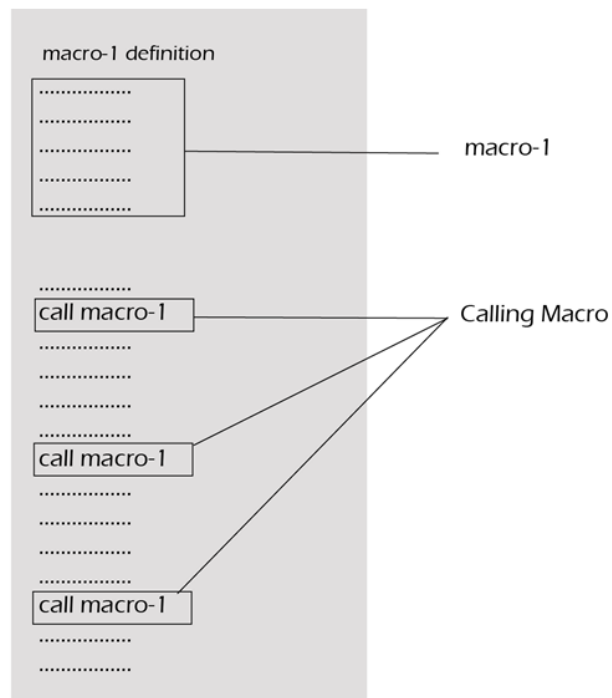
Program Y without macro



برنامه Y دارای بلوک دستورات یکسان در سه مکان مختلف است که یک کد اضافی است. همچنین پیچیدگی برنامه را افزایش می دهد.

نمودار زیر همان برنامه را پس از اجرای Macro نشان می دهد:

Program Y with macro



در نمودار بالا، macro-1 تعریف Macro است و با فراخوانی macro-1 در همه مکان ها جایگزین کد تکراری شده است.

1. DEFINE <macro-name>.
2. .
3. Statements-block
4. .
5. END-OF-DEFINITION.

نحوه صدازدن Macro:

1. <macro-name> [<param1> <param2>....<param9>].

- Macro-name: نام Macro را مشخص می کند.
- Param1, param2,...: پارامترهای Macro را مشخص می کند. پارامترها با نگهدارنده مکان در تعریف کلان جایگزین می شوند. حداکثر تعداد پارامترها/دارندگان مکان مورد استفاده در Macro ۹ عدد است. <param1> <param2>....<param9> جای دارندگان مکان جدا شده با کاما و ۱، &2،، &9 را می گیرد.

مثال:

در مثال زیر، دو گزینه رادیویی (Radio button) داریم که بمحض انتخاب هر کدام از آنها، گزینه انتخاب شده در خروجی نمایش داده خواهد شد. کد مورد نیاز را یکبار بدون استفاده از Macroها و یکبار با استفاده از آنها بنویسید.

کد بدون Macro:

1. *&-----*
2. *& Report Z_MACRO
3. *&-----*
4. *& Written by Beyamooz
5. *&-----*
6. REPORT Z_MACRO.
7. PARAMETERS: OPTION1 type C RADIOBUTTON group RG,
8. OPTION2 type C RADIOBUTTON group RG.
9. START-OF-SELECTION.
10. IF OPTION1 = 'X'.
11. WRITE: 'Macro from option: 1'.
12. SKIP 1.
13. WRITE: 'Radio button selected: 1'.

```

14. ENDF.
15. IF OPTION2 = 'X'.
16.   WRITE: 'Macro from option: 2'.
17.   SKIP 1.
18.   WRITE: 'Radio button selected: 2'.
19. ENDF.

```

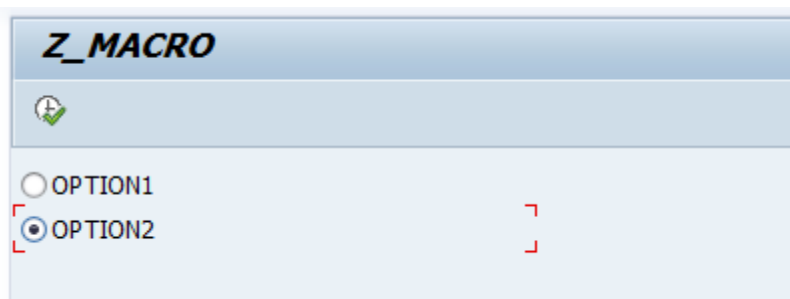
کد با Macro:

```

1.  *&-----*
2.  *& Report Z_MACRO
3.  *&-----*
4.  *& Written by Beyamooz
5.  *&-----*
6.  REPORT Z_MACRO.
7.  DEFINE disp_macro_info.
8.    WRITE: 'Macro call : &1'.
9.    SKIP 1.
10.   WRITE: 'Radio button selected: &1'.
11.  END-OF-DEFINITION.
12.  PARAMETERS: OPTION1 type C RADIOBUTTON group RG,
13.              OPTION2 type C RADIOBUTTON group RG.
14.  START-OF-SELECTION.
15.  IF OPTION1 = 'X'.
16.    disp_macro_info 1.
17.  ENDF.
18.  IF OPTION2 = 'X'.
19.    disp_macro_info 2.
20.  ENDF.

```

خروجی:




```
Z_MACRO  
  
Z_MACRO  
-----  
Macro call : 2  
  
Radio button selected: 2
```

توضیح مثال:

در مثال بالا، قبل از هر دستور، یک Comment برای توضیح در مورد عبارت ارائه شده است. آنها را مرور کنید تا درک واضحی از کد نمونه بدست آورید.

در کد بدون Macro، کد زیر ۲ بار با حداقل تغییرات برای هر دکمه رادیویی، تکرار می شود.

1. WRITE: 'Macro from option: 1'.
2. SKIP 1.
3. WRITE: 'Radio button selected: 1'.

کد را می توان به Macro اضافه کرد و Macro را می توان دو بار برای کاهش افزونگی فراخوانی کرد. بنابراین Macro را می توان مانند زیر نوشت:

1. DEFINE disp_macro_info.
2. WRITE: 'Macro call : &1'.
3. SKIP 1.
4. WRITE: 'Radio button selected: &1'.
5. END-OF-DEFINITION.

صدار زدن Macro همراه با شماره گزینه Radio button

1. disp_macro_info 1

Include ها در ABAP

اگر مجموعه ای از دستورات (سورس کد) در بیش از یک برنامه استفاده شود، آن مجموعه دستورات را می توان تحت قالب یک Include Program با یک نام یکتا، ذخیره نمود و سپس نام Include Program مذکور را در دیگر برنامه ها گنجانده. برای همه برنامه ها در دسترس اند و در هر برنامه ای استفاده می شوند.

Include Program ها فقط برای شکستن برنامه های بزرگ به چند برنامه کوچکتر، استفاده می شوند و نمی توان برای آنها پارامتر در نظر گرفت. Include Program ها برنامه های مستقل نیستند و نمی توانند بطور مستقل اجرا شوند.

سطح دسترسی Include Program ها Global (عمومی) است و در هر برنامه ABAP می توان از آنها استفاده نمود. Include Program شامل قطعه کوچکی از Source Code است که می تواند با دستور INCLUDE در یک برنامه دیگر گنجانده شود.

دستور INCLUDE مسئول کپی مجموعه دستورات Include Program در برنامه اصلی در طول زمان اجراست. Include Program ها نمی توانند خودشان را فراخوانی کنند. Include Program باید عاری از خطای نحوی و حاوی دستورات کامل باشد.

نحوه استفاده:

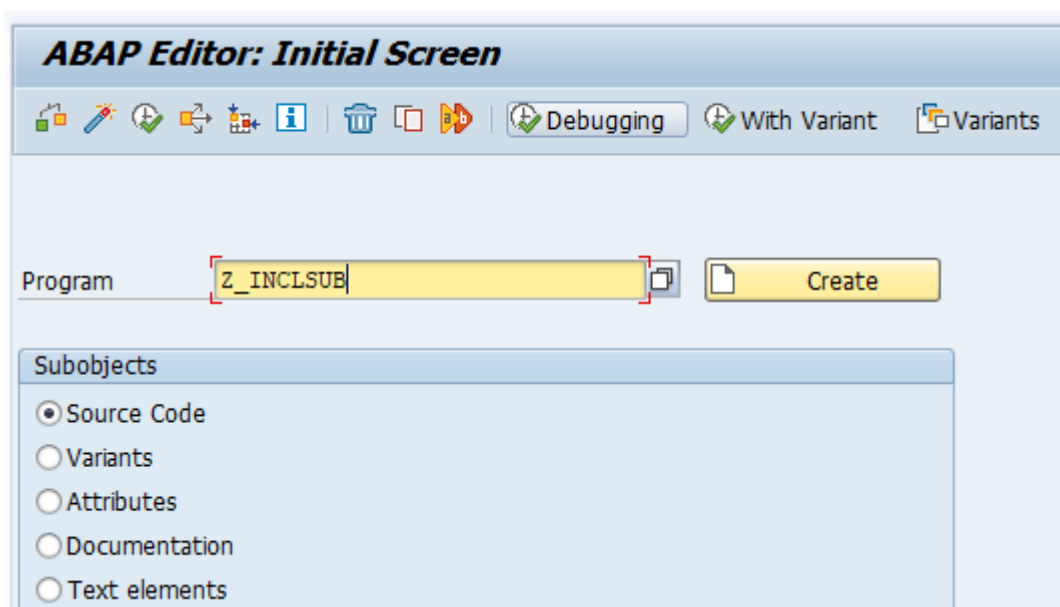
1. INCLUDE <include-program>

- **Include-program**: نام Include Program را مشخص می کند. Include Program ها را می توان در ABAP Editor ایجاد کرد.

مثال:

مثال زیر نحوه ایجاد یک Include Program و نحوه استفاده از آن در برنامه را توضیح می دهد.

مرحله ۱: به تیکد SE38 بروید. نام (Z_INCLSUB) را بعنوان نام Include تنظیم کرده و روی دکمه Create کلیک نمایید.



پایان پیش نمایش این فصل، ادامه در فایل اصلی ...!

توجه: پیش نمایش بالا، تنها با هدف جلب اطمینان شما از محتوای اصلی تهیه شده است. بنابراین اگر مورد تأیید شما قرار گرفت لطفاً از طریق سایت Beyamooz.com اقدام به خرید فایل کامل آن نمایید.

لینک خرید کتاب های آموزش SAP

۱. [کتاب آموزش برنامه نویسی و توسعه SAP | آموزش SAP ABAP \(کتاب پیش رو\)](#)
۲. [کتاب آموزش نرم افزار مدیریت کیفیت SAP | ماژول QM در SAP](#)
۳. [کتاب آموزش پیاده سازی و پشتیبانی SAP | آموزش SAP BASIS](#)
۴. [کتاب آموزش مدیریت پایگاه داده SAP | آموزش SAP HANA](#)
۵. [کتاب آموزش نرم افزار منابع انسانی SAP | ماژول HR در SAP](#)
۶. [کتاب آموزش نرم افزار فروشگاهی SAP | ماژول SD در SAP](#)
۷. [کتاب آموزش حسابداری مدیریت SAP | ماژول CO در SAP](#)
۸. [کتاب آموزش نرم افزار حسابداری SAP | ماژول FI در SAP](#)
۹. [کتاب آموزش نگهداری و تعمیرات SAP | ماژول PM در SAP](#)
۱۰. [کتاب آموزش برنامه ریزی تولید SAP | ماژول PP در SAP](#)
۱۱. [کتاب آموزش انبارداری SAP | ماژول MM در SAP](#)
۱۲. [لیست همه کتاب های SAP](#)

موفق و پاینده باشید

فصل ۱۹: برنامه نویسی شی گرا در ABAP

برنامه نویسی شی گرا چیست؟

در این فصل به بررسی یکی از زیباترین، و در عین حال خطرناکترین مباحث برنامه نویسی می پردازیم، این مبحث از این جهت خطرناک هست که اگر شما اصول اولیه را یاد بگیرید و این نوع برنامه نویسی بر شما تاثیر بگذارد، از آن پس دیگر به همه چیز به چشم یک شیء نگاه خواهید کرد و تمام روشهای برنامه نویسی گذشته خود را در دنیای واقعی کنار خواهید گذاشت.

در اینجا قصد نداریم شروع کنیم به آوردن مثالهایی از Object در دنیای واقعی و فرض می کنیم که شما مثالهایی مثل رنگ ماشین و یا تلفن را بلد باشید! در حالت کلی یک شیء شامل یک سری خصوصیات و توابع می باشد که درون یک قالب کلی به نام Class قرار دارند، به متغیرهای درون کلاسها Attribute و به توابع موجود در آن Method می گویند.

دلیل استفاده از اشیاء در زبان های برنامه نویسی

در برنامه های بزرگ، استفاده از Class ها موجب می شود تا کدهای تکراری نوشته نشوند لذا مقدار کدها کمتر می شود و در نتیجه فشار کمتری به سیستم اجرا کننده دستورات می آید و همچنین سرعت پردازش کدها مقدار زیادی افزایش می یابد.

به عبارت دیگر ویژگی Class ها در این است که می توان یک کلاس را بعداً با نام های مختلف و متغیرهای مختلف برای استفاده های مشابه استفاده نمود.

Object (شیء) چیست؟

مجموعه ای از متغیرها و توابع است که از یک الگوی خاص به نام Class ساخته شده است، Object (شیء) می گویند. اما Class ها چه هستند؟ فرض کنید ما یک شرکت داریم که این شرکت از بخش های مختلفی تشکیل شده است حال ما در هر بخش احتیاج داریم که هر ماه یک گزارش مالی بگیریم! اما اینجا دو کار می توانیم بکنیم هم می توانیم برای هر بخش چند نفر بزاریم و آنها گزارش مالی را تهیه کنند و هر ماه تحویل دهند در این صورت در هر بخش شلوغی و همچنین کارمند بیشتری نیازمندیم! راه دیگر اینست که یک قسمت به عنوان اتاق گزارش مالی درست کنیم و هر بخش داده های خود را به این قسمت بدهد و گزارش مالی خود را دریافت کند در این روش هم بخش ها منظم تر خواهند بود و هم دیگر احتیاج به کارمند اضافی نداریم.

در اینجا Object ها، نقش کارمند در بخش گزارش مالی را بر عهده دارند.

Class چیست؟

یک Class مجموعه ای از Attribute (ویژگی) ها و Method (دستورالعمل) ها است. Class یک الگو برای تعریف اشیاء است. از نظر فنی، اشیاء Instance (نمونه) های زمان اجرای یک Class هستند.

هر تعداد شی را می توان روی یک Class ایجاد کرد. هر Instance (نمونه) از یک Class دارای هویت یکتا و مجموعه ای از مقادیر خاص برای Attribute های خود است.

انواع Class ها در ABAP

Class های ABAP بر اساس اعلان و قابل مشاهده بودن به دو نوع زیر تقسیم می شوند:

| Class | توضیحات |
|------------------------------------|--|
| Global Classes (کلاس های عمومی) | Global Class ها توسط همه برنامه های ABAP قابل دسترس هستند. با استفاده از تیکد SE24 در Workbench ABAP (میزکار ABAP) می توان آنها را ایجاد نمود. این نوع کلاس ها بصورت مرکزی، در class pools -> class library -> Repository ذخیره می شوند. |
| Local Classes (کلاس های محلی) | Local Class ها فقط در همان برنامه ای که تعریف شده اند، قابل دسترس و استفاده هستند و برای دیگر برنامه ها در دسترس نیستند. این نوع کلاس ها از Source Code تشکیل شده اند و باید بین CLASS و ENDClass کد شوند. |

هنگامی که یک Class در یک برنامه ABAP وجود دارد، سیستم ابتدا یک Local Class را با نام مشخص شده جستجو می کند. اگر هیچ Local Class ای پیدا نکرد، به دنبال یک Global Class می گردد. جدا از قابلیت دسترسی، هیچ تفاوت دیگری بین استفاده از یک Global Class یا یک Local Class وجود ندارد.

نحوه تعریف Class در ABAP

تعریف Class شامل ۲ بخش زیر است:

۱- Declaration Part (بخش اعلان): این بخش شامل اعلان تمام اجزاء Class از جمله Attribute ها، Method ها و

Event هاست. با استفاده از دستور CLASS <class-name> DEFINITION این بخش تعریف می شود.

۲- Implementation Part (بخش پیاده سازی): این بخش شامل پیاده سازی Method هایی است که در مرحله قبل

اعلان شدند. با استفاده از دستور CLASS <class-name> IMPLEMENTATION این بخش تعریف می شود.

نحوه ی Declaration (اعلان) یک Class:

1. CLASS <class-name> DEFINITION.
2. ...
3. Statements-block
4. ...
5. ENDClass.

دستور بالا شامل اعلان برای تمام اجزای Class (Attribute ها، Method ها، Event ها) است. توصیه می شود، اعلان Local Class را در ابتدای برنامه قرار دهید زیرا این قسمت متعلق به اعلان داده های برنامه است.

نحوه ی Implementation (پیاده سازی) یک Class:

1. CLASS <class-name> IMPLEMENTATION.
2. ...
3. Statements-block
4. ...
5. ENDCLASS.

قسمت پیاده سازی Class شامل Source Code تمام Method های Class است.

اجزاء تشکیل دهنده Class

اجزاء Class محتویات آنرا تشکیل می دهند. همه اجزاء Class در Declaration Part (بخش اعلان) اعلان می شوند. اجزاء مختلف Class، خصوصیات و رفتار اشیاء ساخته شده از آنرا تعریف می کنند. می توان به تمام اجزاء در یک Class دسترسی داشت. به این اجزاء Component نیز می گویند.

دو نوع Component (جزء) برای Class وجود دارد:

۱- Instance Components (اجزاء مختص نمونه): برای هر شیء ساخته شده از Class بطور جداگانه وجود دارند و در واقع مختص نمونه هستند.

۲- Static Components (اجزاء استاتیکی): فقط یک بار برای کل Class بدون توجه به تعداد نمونه ها بوجود می آیند و در واقع برای همه نمونه های ساخته شده از Class یکسان هستند.

در جدول زیر، اجزاء مختلف Class را بررسی کرده ایم:

| Class Components (اجزاء کلاس) | توضیحات |
|----------------------------------|--|
| Attributes (خصوصیات) | Attribute ها فیلدهای داخلی یک Class هستند که با هر نوع داده ABAP مانند C، A، F و N اعلان می شوند. وضعیت یک شی با مقدار Attribute های آن تعیین می شود. |
| Methods (متدها) | Method ها رویه های داخلی یک Class هستند و رفتار یک شی را تعریف می کنند. Method ها می توانند به تمام Attribute های Class دسترسی داشته باشند و مقدار آنها را تغییر دهند. |
| Constructors (سازنده ها) | Constructor ها متدهای خاصی هستند که بطور خودکار هنگام ایجاد یک شی یا زمان ایجاد نمونه از Class فراخوانی می شوند. Constructor هنگامی که یک شی ایجاد می شود فعال می شود. |

| | |
|------------------------|---|
| Events (رویدادها) | Event ها توابعی هستند که بر اساس نتیجه یک شرط فعال می شوند. Event ها توسط Object ها یا Class ها برای راه اندازی Method های کنترل کننده رویداد در اشیاء یا Class های دیگر استفاده می شوند. |
| Types (نوع ها) | در یک Class با استفاده از دستور Type می توان Data Type های دلخواهتان را تعریف کنیم. به این Data Type های تعریف شده توسط کاربر "User-Defined" می گوئیم. Type ها برای تمام اشیاء ساخته شده از یک کلاس فقط یک بار بوجود می آیند. |
| Constants (ثابت ها) | Constant ها در واقع Attribute هایی هستند که مقدارشان ثابت است و با عبارت CONSTANTS می توان آنها را اعلان نمود. Constant ها مختص یک نمونه نیستند و یکبار برای همه اشیاء Class بوجود می آیند. |

پایان پیش نمایش این فصل، ادامه در فایل اصلی ...!

توجه: پیش نمایش بالا، تنها با هدف جلب اطمینان شما از محتوای اصلی تهیه شده است. بنابراین اگر مورد تأیید شما قرار گرفت لطفاً از


طریق سایت Beyamooz.com اقدام به خرید فایل کامل آن نمایید.

لینک خرید کتاب های آموزش SAP

۱. [کتاب آموزش برنامه نویسی و توسعه SAP | آموزش SAP ABAP \(کتاب پیش رو\)](#)
۲. [کتاب آموزش نرم افزار مدیریت کیفیت SAP | ماژول QM در SAP](#)
۳. [کتاب آموزش پیاده سازی و پشتیبانی SAP | آموزش SAP BASIS](#)
۴. [کتاب آموزش مدیریت پایگاه داده SAP | آموزش SAP HANA](#)
۵. [کتاب آموزش نرم افزار منابع انسانی SAP | ماژول HR در SAP](#)
۶. [کتاب آموزش نرم افزار فروشگاه SAP | ماژول SD در SAP](#)
۷. [کتاب آموزش حسابداری مدیریت SAP | ماژول CO در SAP](#)
۸. [کتاب آموزش نرم افزار حسابداری SAP | ماژول FI در SAP](#)
۹. [کتاب آموزش نگهداری و تعمیرات SAP | ماژول PM SAP](#)
۱۰. [کتاب آموزش برنامه ریزی تولید SAP | ماژول PP SAP](#)
۱۱. [کتاب آموزش انبارداری SAP | ماژول MM در SAP](#)
۱۲. [لیست همه کتاب های SAP](#)

موفق و پاینده باشید

همراهان خوبی باشید و با

ری پست کردن ( Repost) این محتوی در [لینکدین پیاموز](#)
ما را در تولید دوره های آموزشی مفید
حمایت کنید.

از طریق صفحه لینکدین پیاموز، ما را **فالو** کنید.

+ Follow

برای باخبر شدن از آخرین دوره های آموزشی، کافیه در
لینکدین پیاموز روی **آیکن زنگ** کلیک کنید.



Beyamooz

[صفحه پیاموز در لینکدین](#) 