

دوره آموزشی جاوا استاندارد



طول دوره: ۲۰ جلسه (در مجموع ۴۹ ساعت)
مدرس: احمد رضا صدیقی



سرفصل‌های دوره آموزشی جاوا استاندارد

۱. مقدمه و آشنایی با جاوا

- تاریخچه جاوا
- چگونگی کارکرد جاوا
- خانواده جاوا
- معرفی JVM، JRE، و JDK
- آماده کردن محیط برنامه نویسی جاوا در ویندوز و لینوکس
- نوشتن اولین برنامه جاوا، کامپایل و اجرا
- پنجره دستور و ورودی و خروجی استاندارد
- پیاده سازی چندین متد
- پیاده سازی چندین کلاس
- نصب IntelliJ IDEA

در این جلسه، با چگونگی کارکرد جاوا و تفاوت‌های جاوا با زبان‌های محلی (native) مثل C و ++C آشنا می‌شوید و فرایند «کامپایل» و «اجرای» برنامه‌های جاوا را می‌آموزید. همچنین می‌آموزید که چگونه محیط برنامه نویسی جاوا را در ویندوز و لینوکس آماده کنید و یک برنامه ساده به زبان جاوا بنویسید و اجرا کنید. در انتها نیز با نصب IntelliJ IDEA و استفاده از آن به عنوان ویرایشگر و محیط برنامه نویسی (IDE) جاوا آشنا می‌شوید.

۲. مفاهیم پایه

- انواع داده ها
- متغیرها و قوانین نامگذاری آنها
- محدوده قابل مشاهده متغیرها
- عملگرها (محاسباتی، مقایسه ای، منطقی، بیتی، انتساب)
- ساختارهای کنترلی (if، if-else، for، while، do-while، switch-case)
- کلمات کلیدی return و continue
- آرایه ها

در زبان جاوا انواع داده‌ها قابل استفاده اند. داده‌های عددی صحیح، عددی اعشاری، و کاراکتر از جمله این داده‌ها هستند. روی این داده‌ها می‌توانید مجموعه‌ای از عملیات را انجام دهید که انجام این عملیات با استفاده از عملگرهای مختلف این زبان (محاسباتی، مقایسه ای، منطقی، بیتی، انتساب) قابل انجام هستند.

در جاوا از ساختارهای کنترلی برای کنترل اجرای برنامه، بررسی شرایط، یا تکرار عملیات استفاده می‌شود. در پایان، با آرایه‌های دو بعدی و چندبعدی و نحوه تعریف و مقداردهی آنها در زبان جاوا آشنا می‌شوید.

۳. شی گرایی

- کلاس به عنوان داده جدید
- افزودن متد به کلاس
- سازنده
- finalize()
- متدهای toString() و equals()
- کلمه کلیدی this
- Overloading
- تعریف پکیج (package)
- فیلدهای static و final

شی گرایی روش اصلی در توسعه اکثر قریب به اتفاق نرم افزارهای امروزی است. اهمیت شی گرایی به اندازه ای است که اگر یک برنامه نویس، شی گرایی را نداند می توان گفت او برنامه نویس نیست. متأسفانه بسیاری از برنامه نویسان امروزی درک دقیق و کاملی از شی گرایی ندارند و شی گرایی را به صورت کلی و غیر دقیق بلد هستند. در این جلسه، علاوه بر مفاهیم شی گرایی، کاربرد و موارد استفاده هر یک از مفاهیم و جزئیات شی گرایی با مثال‌هایی کاربردی و عملی به شما آموخته می‌شود تا پس از این جلسه درک دقیق و کاملی از شی گرایی به دست می‌آورد.

۴. ارث بری

- ارث بری بین کلاسها، چرا و چگونه؟
- دسترسی به کلاس پدر با استفاده از super
- Overriding
- پیاده سازی سازنده در کلاس فرزند
- کلاس abstract
- کلاس final
- کلاس java.lang.Object

ایده ارث بری از دنیای واقعی گرفته شده است، زیرا موجودات دنیای واقعی، شباهت‌های زیادی به هم دارند به گونه‌ای که می‌توان تصور کرد این خصوصیات را از یکدیگر به ارث برده‌اند. از آنجاییکه هدف شی گرایی، شبیه سازی دنیای واقعی است، ارث بری به شی گرایی نیز راه پیدا کرده است. در این جلسه، با ارث بری و

اینکه چگونه به ما در طراحی، تولید و نگهداری برنامه کمک می‌کند آشنا می‌شوید. مثل همیشه، مثال‌های کاربردی و عملیاتی را برای توضیح و پیاده سازی ارث بری ملاحظه خواهید کرد.

۵. اینترفیس و Enumeration

- اینترفیس چیست و چگونه تعریف می‌شود
- پیاده سازی اینترفیس
- چند نکته در رابطه با اینترفیس‌ها
- ارث بری میان اینترفیس‌ها
- Enum چیست و چرا؟
- جزئیات پیاده سازی Enumها

اینترفیس‌ها یکی دیگر از مفاهیم شی گرای است که طراحی برنامه‌های جاوا را ساده و انعطاف پذیر می‌کند. Enum مکانیسمی برای تعریف مقادیر ثابت در برنامه‌های جاوا هستند که در این جلسه با آن آشنا می‌شوید.

۶. کنترل خطا و استثنا

- مفهوم استثنا
- استفاده از Exception در کنترل خطا و استثنا
- توسعه کلاس Exception
- تولید بیش از یک Exception در یک متد
- بلوک finally
- اینترفیس AutoClosable
- متدهای کلاس Exception
- تولید خطاهای زنجیره‌ای
- متدهای (GetMessage)، (GetCause) و (PrintStackTrace)
- استثنای Runtime و Non-Runtime

در تولید برنامه‌ها، همواره سناریوهای خوشبینانه مدنظر هستند. اما وقتی یک برنامه در محیط عملیاتی قرار می‌گیرد، شرایط محیطی، وضعیت داده‌ها، نحوه تعامل یک کاربر با برنامه منجر به شرایط و استثنای مختلفی می‌شوند که ممکن است روال اجرای برنامه را از وضعیت نرمال خارج کنند. در چنین شرایطی برنامه می‌بایست شرایط غیر نرمال را کنترل کند و عکس العمل مناسب نشان دهد. کنترل استثنا و خطا بخش مهمی از برنامه نویسی جاوا و زبان‌های دیگر است و بدون آن، تولید هیچ برنامه‌ای (ساده یا پیچیده) غیرممکن است. طراحی درست و استفاده بجای کنترل خطا و استثنا، توسط بسیاری از برنامه نویسان انجام نمی‌شود و در این جلسه شما با اصول و روش صحیح پیاده سازی آن آشنا خواهید شد.

۷. پکیج java.lang

- کلاس‌های Wrapper (Integer، Double، Long، Character، ...)
- کلاس System
- کلاس Object
- کلاس Cloneable
- کلاس Class
- کلاس Math

مهمترین بخش از جاوا را مجموعه‌ای از کلاس‌ها تشکیل می‌دهند که در این جلسه به آنها پرداخته می‌شود.

۸. کاراکترها و رشته‌ها

- کلاس String و سازنده‌های آن
- متصل کردن رشته‌ها
- متدهای کلاس String
- مقایسه کردن Stringها
- مکانیابی درون رشته‌ها
- استخراج رشته‌های زیرمجموعه یک رشته
- کلاس StringBuffer

کار با رشته‌ها و متن‌ها نیاز هر برنامه است. داده‌هایی که توسط کاربر وارد می‌شوند، داده‌هایی که در پایگاه داده ذخیره و بازیابی می‌شوند، یا داده‌هایی که از یک فایل متنی خوانده می‌شوند یا در یک فایل متنی نوشته می‌شوند، همگی متن و رشته هستند و برنامه باید بتواند آنها را نگهداری و پردازش کند. در این جلسه با نحوه تعامل با رشته‌ها آشنا می‌شوید

۹. ساختمان داده‌ها و پکیج java.util

- Collectionها
- اینترفیس Collection
- اینترفیس List
- اینترفیس Set
- اینترفیس SortedSet
- اینترفیس Queue
- کلاسهای Collection
- کلاس ArrayList
- کلاس LinkedList

- کلاس HashSet
- اینترفیس Iterator
- حلقه For
- Map
- اینترفیس Map
- اینترفیس SortedMap
- اینترفیس Map.Entry
- کلاس های Map
- کلاس HashMap
- کلاس Arrays
- کلاس و اینترفیس های قدیمی در پکیج java.util
- اینترفیس Enumeration
- کلاس Vector
- کلاس Stack
- کلاس Hashtable
- کلاس Properties
- کلاس Date
- کلاس Calendar
- کلاس Random

در کار با داده‌ها، موارد فراوانی وجود دارد که نیاز است داده‌ها را جستجو کنیم، یا آنها را مرتب کنیم، یا داده‌های تکرار را حذف کنیم یا به سرعت به آنها دسترسی پیدا کنیم. به این منظور مجموعه‌ای از ساختمان داده‌های آماده در زبان جاوا پیاده سازی و آماده شده است که در این فصل با آنها آشنا می‌شوید. ساختمان داده‌های معروفی از قبیل List (که داده‌های خود را به ترتیب نگهداری می‌کند) یا Set (که داده‌های غیرتکراری را نگهداری می‌کند) از معروف ترین اینها هستند. در این جلسه همچنین با کلاس‌های Date و Calendar که برای کار با تاریخ و زمان استفاده می‌شوند آشنا می‌شوید.

۱. Generics

- Generics چرا و چگونه؟
- نامگذاری پارامترهای Generics
- متدها و سازنده‌های Generics
- استفاده از نشانه‌های extends و super؟
- ارث بری

یکی از قابلیت‌هایی که انعطاف پذیری کدهای جاوا و استفاده مجدد (Reusability) آنها را افزایش می‌دهد مفهوم Generics است که در این جلسه با ارایه مثال‌های کاربردی و عملی آشنا می‌شوید

۱۱. ورودی و خروجی (File/Stream)

- کار با فایل‌ها و دایرکتوری‌ها با استفاده از کلاس File
- خواندن یا نوشتن در فایل‌ها با استفاده از Streamها
- بافر کردن داده‌ها به منظور کارایی بالاتر
- Serialization
- کلاس Scanner

امروزه تقریباً در تمام برنامه‌ها با فایل‌ها سروکار داریم. ممکن است تنظیمات یک برنامه را در یک فایل ذخیره کنیم، یا بخواهیم فایل‌های عکس، فیلم یا صوت را در برنامه بارگذاری و نمایش دهیم. در این فصل چگونگی کار ایجاد فایل یا دایرکتوری، خواندن فایل‌ها یا نوشتن در آنها را خواهید آموخت. علاوه بر فایل‌ها، خواندن یا نوشتن داده‌ها را در منابع مختلف (مثلاً حافظه سیستم) یاد می‌گیرید. در انتها با نحوه سریال کردن داده‌های جاوا برای عبور از شبکه یا ذخیره در فایل را خواهید آموخت.

۱۲. Threadها و همزمانی

- موازی سازی بخش‌های برنامه
- کنترل همزمانی با استفاده از Synchronization
- اجرا کننده‌ها
- اینترفیس Callable
- اینترفیس Future
- کلاس Timer و TimerTask

یکی دیگر از مفاهیم مهم در برنامه‌های امروزی، موازی سازی اجرای بخش‌هایی از برنامه است تا به این ترتیب کارایی برنامه و رضایت کاربر افزایش یابد. تصور کنید در زمانیکه کاربر یک فایل را آپلود می‌کند (که زمان زیادی صرف می‌کند)، برنامه می‌تواند به صورت همزمان کار دیگری را نیز انجام دهد. در این فصل استفاده از Threadها و مفاهیم پیاده سازی آنها را در مثال‌های عملی و کاربردی ملاحظه خواهید کرد و چالش‌ها آنها را خواهید آموخت.

ارتباط با پایگاه داده (JDBC)

- آشنایی با SQL
- مفاهیم JDBC (Statement, Connection, Driver)
- جستجو در پایگاه داده (دستور select و join)

- بروزرسانی پایگاه داده (delete, update, insert)
- مدیریت تراکنش
- کلاس Entity

کار با پایگاه داده‌ها، بخش اصلی و همیشگی برنامه‌های امروزی است. بدون پایگاه داده، تقریباً هیچ برنامه‌ای امروزه تولید نمی‌شود! در این جلسه یاد می‌گیرید که چگونه در یک برنامه جاوا به پایگاه داده متصل شوید، داده‌هایی را در آن ذخیره یا بارگذاری کنید. نگران نباشید، برای کار با پایگاه داده نگاهی اجمالی به SQL و مفاهیم جدول، ستون، کلید اصلی، و کلید خارجی خواهیم انداخت، همچنین مفهوم مهم «مدیریت تراکنش» و چگونگی انجام آن در برنامه‌های جاوا را یاد خواهید گرفت.

عبارت‌های باقاعده (Regular Expression)

- تعریف قاعده برای رشته‌ها
- کاراکترهای خاص
- محدوده کاراکترها
- میانبرها
- تکرار
- گروه بندی کاراکترها
- ارجاع به عقب
- انطباق در نقاط مرزی
- کلاس Pattern
- کلاس Matcher
- کلاس PatternSyntaxException

رشته‌هایی که در برنامه‌ها پردازش می‌شوند ممکن است همگی یک قالب داشته باشند، مثلاً آدرس وب سایت‌ها، ایمیل‌ها، شماره تلفن‌ها، کد پستی‌ها، همگی یک قالب دارند، به این قالب‌ها «عبارت‌های باقاعده» گفته می‌شود. برنامه‌ها باید بتوانند انطباق یک رشته را بر یک عبارت با قاعده بررسی کنند، یا در یک رشته جستجو کنند و تمام رشته‌های منطبق بر یک عبارت را بیابند. در این فصل کار با عبارت‌های باقاعده که اهمیت زیادی در کار با رشته‌ها دارند را خواهید آموخت.

۱۳. Annotation

- Annotation چرا و چگونه
- انواع Annotation
- عناصر Annotation
- محدوده قابل استفاده از یک Annotation

• سیاست‌های ماندگاری

Annotation ها نوعی علامتگذاری در کدهای برنامه هستند، علامت‌هایی که توسط جاوا فهمیده و پردازش می‌شوند. امروزه تمام Java Enterprise مبتنی بر Annotation هاست، بنابراین درک و نحوه پیاده سازی Annotation ها اهمیت بسیاری در یادگیری Java Enterprise دارد، البته اگر قصد یادگیری Java Enterprise را داشته باشید!

۱۴. واسط کاربری (Swing)

- کلاس JFrame
- چیدمان نمایش اجزای واسط کاربری (Layout Management)
- کامپوننت‌های واسط کاربری
- منوها

واسط کاربری بخش مهمی از اغلب برنامه‌هاست. تکنولوژی Swing یکی از تکنولوژی‌های جاوا استاندارد برای تولید واسط کاربری است که در این جلسه جزئیات آنرا خواهید آموخت.

تکنولوژی Swing یک خصوصیت ممتاز به نام «Look And Feel» دارد که امکان می‌دهد تیم یا پوسته نمایش واسط کاربری را بدون تغییر در کدهای برنامه تغییر دهیم. به این ترتیب، وقتی واسط کاربری یک برنامه را توسعه می‌دهید بدون اینکه به شمایل یا زیبایی واسط کاربری فکر کنید، صرفاً کارکرد آنرا پیاده سازی و تست می‌کنید، بعداً می‌توانید هر پوسته‌ای بخواهید برای آن انتخاب کنید.

یک نکته جالب دیگر در مورد واسط کاربری Swing این است، که پنجره‌ها، منوها، دیالوگ‌ها و تمام اجزای واسط کاربری در لینوکس، ویندوز یا مک یکسان و به یک شکل نمایش داده می‌شود.

۱۵. Logging

- چیست Logging
- آجکت Logger
- سطوح لاگ
- آجکت Handler
- الگوی نامگذاری فایل لاگ
- آجکت Formatter
- آجکت LogManager

در برنامه های عملیاتی که با داده‌های مهم و ارزشمند کار می‌کنند، ردیابی تمام عملیات برنامه، (ورود و خروج کاربران، ثبت داده، ...) حوادث و رخدادهای برنامه (خطاها و استثناها)، و تغییرات داده (تغییر رمز کاربر، ویرایش پروفایل، ...) اهمیت زیادی دارد. Logging یک مکانیسم کارآمد و بسیار مناسب برای ثبت رویدادها و عملیات مختلف برنامه است. با استفاده از Logging، می‌توانید تمام حوادث برنامه را بدون اینکه تاثیر قابل

توجهی در کارایی و کارکرد برنامه داشته باشد در فایل‌های متنی که به آنها فایل‌های لاگ گفته می‌شود ذخیره کنید تا بعداً به آنها مراجعه کنید و روند حوادث را در صورت نیاز بررسی کنید.

عبارت‌های Lambda و کلاس‌های تودرتو

- کلاس داخلی
- کلاس داخلی استاتیک
- کلاس داخلی غیر-استاتیک
- کلاس محلی
- کار با کلاس بیرونی
- کلاس بی نام (Anonymous)
- Listener
- عبارت‌های لامبدا (Lambda)
- اینترفیس Functional
- کلاس Stream
- متدهای پیش فرض و متدهای استاتیک
- قواعد متدهای پیش فرض
- متدهای استاتیک

مدل سازی بسیاری از موجودات دنیای واقعی در نرم افزار صرفاً با استفاده از مفاهیم پایه در شی گرای دشار است به همین دلیل جاوا، امکان تعریف یک کلاس داخل یک کلاس دیگر را امکانپذیر کرده است. در این جلسه، کاربرد و اهمیت چنین قابلیت‌هایی را با چندین مثال کاربردی بحث خواهیم کرد و سپس استفاده از عبارت‌های لامبدا در «برنامه نویسی رویه‌ای» را خواهید آموخت. در انتها، با متدهای پیش فرض و استاتیک و کاربرد آنها آشنا خواهید شد

۱۶. Reflection

- واکاوی جزئیات کلاس‌ها
- دسترسی به اجزای داخلی کلاس‌ها (فیلدها، متدها و سازنده‌ها)
- مثال کاربردی

Reflection یکی از تکنولوژی‌های جذاب زبان جاواست که امکان می‌دهد با استفاده از برنامه نویسی، جزئیات کلاس‌ها را واکاوی کنیم، آنها را دستکاری کنیم یا در زمان اجرای برنامه آنها را فراخوانی کنیم. قسمت عمده‌ای از Java EE مبتنی بر Reflection است. دانستن Reflection به درک و یادگیری مفاهیم پیشرفته جاوا کمک می‌کند.

چندزبانی

- فایل‌های Resource Bundle
- آبجکت‌های Locale
- واکنشی متن از آبجکت ResourceBundle
- متن‌ها و پیام‌ها
- فرمت دهی
- استفاده از فرمت‌های آماده
- فرمت‌های دلخواه برای تاریخ و زمان
- تغییر نمادهای تاریخ و زمان
- پیام‌ها

بسیاری از نرم افزارهایی که امروزه تولید می‌شوند به بیش از یک زبان (فارسی، عربی، انگلیسی،...) کار می‌کنند. کاربر می‌تواند در صفحه اصلی برنامه زبان مورد نظر خود را انتخاب کند، و تا آخر با زبانی که انتخاب کرده با برنامه کار کند. متن‌های صفحات، چیدمان صفحات (راست-چین یا چپ-چین)، تقویم صفحات، واحد پول،... مطابق با زبانی که کاربر انتخاب می‌کند به کاربر نمایش داده خواهد شد. نکته جالب این است که برای تمام زبان‌ها فقط یک کد اجرا می‌شود، این خاصیت در این جلسه آموزش داده می‌شود.

۱۷. ابزار JAR

- JAR، مکانیسم بسته بندی برنامه‌های جاوا
- فایل Manifest
- اجرایی کردن فایل JAR

وقتی یک برنامه جاوا توسعه داده و تولید می‌شود، مجموعه‌ای از فایل‌ها بوجود می‌آیند. ولی وجود چندین (صدها و در بعضی مواقع هزاران) فایل که در واقع بخش‌های یک برنامه هستند می‌تواند کار نگهداری و حمل و نقل آنها را دشوار کند. ابزار JAR امکان می‌دهد تا فایل‌های یک برنامه را در قالب یک فایل (با پسوند .jar) و

تجمیع کنیم. فایل JAR مشابه یک فایل ZIP است که خود از مجموعه‌ای از فایل‌ها و دایرکتوری‌ها تشکیل شده است. در این جلسه جزئیات ایجاد یک فایل JAR و اجرایی کردن آنرا خواهید آموخت.

۱۸. Javadoc

- Javadoc چیست
- علامت‌های قابل استفاده در javadoc
- تولید مستندات از روی javadoc

مستندات سازی یکی از بخش‌های با اهمیت نرم افزار است. اما در بسیاری از مواقع وظیفه برنامه نویسان نیست! Javadoc نوعی از مستندات است که توسط برنامه نویسان لابلای کدها نوشته می‌شود. این مستندات مشابه توضیحات (comment) لابلای کدها نوشته می‌شوند ولی توسط یک ابزار با نام javadoc (یکی از ابزارهای JDK) قابل شناسایی هستند. ابزار javadoc می‌تواند مستندات Javadoc که داخل کدهای برنامه قرار دارند را استخراج کند و از روی آنها فایل‌ها HTML تولید کند.

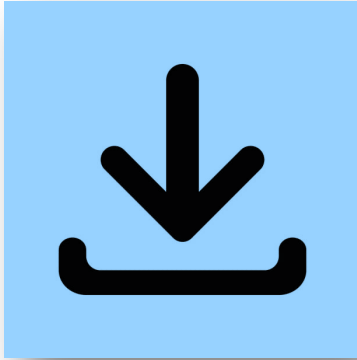
مدرس این دوره احمدرضا صدیقی کیست؟



- معمار ارشد در حوزه جاوا مربوط به پروژه دانشگاه علوم پزشکی
- معمار ارشد در حوزه جاوا مربوط به پروژه شرکت خیره پردا
- معمار ارشد در حوزه جاوا مربوط به پروژه شرکت کیاتک بنیا
- معمار ارشد در حوزه جاوا مربوط به پروژه دانشگاه مالک اشتر
- مشاور پروژه ملی طرح جامع مالیاتی
- مشاور پروژه ملی وزارت بهداشت
- مشاور پروژه بانک ملت
- مولف مجموعه کتابهای جاوا (فارسی و انگلیسی)
- بیش از ۱۲ سال سابقه تدریس جاوا
- ارائه فریمورک تخصصی جاوا (اطلس)

دوره در یک نگاه کلی:

- عنوان دوره:** دوره آموزش برنامه نویسی وب با جاوا
- مخاطبین:** افرادی که می‌خواهند برنامه نویسی وب با جاوا را یاد بگیرند.
- پیش‌نیاز دوره:** دوره جاوا استاندارد
- طول دوره:** ۴۶ ساعت
- پشتیبانی:** دارای گروه تلگرامی اختصاصی است
- نحوه ارائه:** غیرحضوری و داندودی
- نحوه داندود:** داندود از طریق پنل کاربری
- مدیریت دوره:** فرید طاهری مدیریت نیک آموز
- مدرس:** احمدرضا صدیقی



نحوه دریافت دوره‌های دانلودی چگونه است؟

پس از ثبت سفارش، به حساب کاربری خود در سایت نیک‌آموز وارد شده و در بخش «دانلودها» اقدام به دانلود جلسات دوره خریداری شده کنید.



صدور فاکتور رسمی چگونه است؟

در صورت تمایل به دریافت فاکتور رسمی، پیش از خرید خود با واحد فروش مجموعه (۰۲۱ - ۹۱ ۰۷ ۰۰ ۱۷) تماس حاصل نمایید. شایان ذکر است، امکان صدور فاکتور رسمی پس از خرید آنلاین از سایت مجموعه به هیچ عنوان وجود نخواهد داشت.



پشتیبانی بی نظیر

آموزش بدون پشتیبانی کاملا بی‌معنی است، الان تمام دوره‌های نیک آموز دارای پشتیبانی از طریق سایت و تلگرام است.

پس ثبت و نهایی شدن سفارش شما در سایت نیک‌آموز، تیم پشتیبانی طی ۲۴ تا ۷۲ ساعت کاری با شما تماس خواهند گرفت تا فرایند عضو شدن شما در گروه پشتیبانی تلگرامی هر دوره انجام شود. در صورت وجود هر گونه سوال و ابهامی می‌توانید با شماره‌های شرکت تماس حاصل فرمایید و یا از طریق بخش چت پشتیبانی سایت، سوالات خود را مطرح نمایید.

آدرس: تهران، یوسف آباد، میدان فرهنگ، خیابان ۳۳، پلاک ۲۹، زنگ ۲، دفتر نیک آموز
شماره تماس: ۰۲۱ - ۹۱ ۰۷ ۰۰ ۱۷ | **موبایل فروش:** ۰۹۱ ۴۰۰۶ ۲۰۶