



# ainfct

المؤسسة العربية للاستشارات والتدريب وتطوير الأداء  
AINFCT | info@ainfct.com | www.ainfct.com

## Java Programming Fundamentals Course

### فكرة الدورة

تُعد Java من أكثر لغات البرمجة استخداماً في بناء التطبيقات المؤسسية، والخدمات الخلفية، وتطبيقات سطح المكتب، والأنظمة متعددة المنصات. ومع استمرار اعتماد المؤسسات على حلول رقمية قابلة للصيانة والتوسع، يصبح امتلاك أساس برمجي متين عاملاً مهماً للدخول إلى مسارات تطوير البرمجيات الحديثة.

يركز هذا البرنامج التدريبي من AINFCT على تمكين المشاركين من فهم مبادئ البرمجة بلغة Java بدءاً من إعداد بيئة التطوير وكتابة التعليمات الأساسية، وصولاً إلى البرمجة الكائنية، ومعالجة الأخطاء، والتعامل مع الملفات والمجموعات، وبناء تطبيق تدريبي مصغر. ويجمع البرنامج بين الشرح المفاهيمي، والتمارين العملية، والمراجعة التدريجية للكود.

يوفر البرنامج مساراً تطبيقياً واضحاً يساعد المشاركين على الانتقال من الفهم النظري إلى كتابة برامج قابلة للقراءة والتنظيم، بما يدعم نموهم المهني في تطوير البرمجيات.

### أهداف الدورة

فيما يلي الأهداف الرئيسية لهذا البرنامج التدريبي:

- فهم أساسيات لغة Java وبيئة تشغيلها.
- كتابة برامج بسيطة باستخدام تراكيب Java الأساسية.
- تطبيق مبادئ البرمجة الكائنية في نماذج عملية.
- معالجة الأخطاء والملفات والمجموعات بفاعلية.
- بناء تطبيق Java مصغر وفق تنظيم واضح.
- استخدام ممارسات كود نظيف واختبار أساسي.

## منهجية الدورة

- شرح تفاعلي مدعوم بأمثلة كود قصيرة.
- تمارين عملية يومية على المفاهيم الأساسية.
- مراجعة جماعية للأخطاء البرمجية الشائعة.
- تطبيق مشروع تدريبي مصغر خلال البرنامج.
- توجيه عملي لتحسين تنظيم الكود وقراءته.

## أثر الدورة على المنظمة

- يمكن تعزيز قدرات تطوير البرمجيات المؤسسية من خلال:
- رفع جاهزية الفرق للمشاركة في مشاريع Java.
  - تحسين جودة الكود الأساسي داخل التطبيقات.
  - تقليل أخطاء التنفيذ الناتجة عن ضعف الأساسيات.
  - دعم مسارات تأهيل مطوري البرمجيات الجدد.

## أثر الدورة على المتدربين

يساعد البرنامج المشاركين على:

- اكتساب أساس عملي في برمجة Java.
- فهم منطق البرمجة الكائنية بوضوح.
- حل مشكلات برمجية بسيطة بطريقة منظمة.
- الاستعداد لمسارات تطوير Java المتقدمة.

## الشهادات

شهادة معتمدة من AINFCT

## الفئة المستهدفة

يناسب هذا البرنامج المشاركين الراغبين في دخول مجال تطوير البرمجيات باستخدام Java. كما يفيد العاملين التقنيين الذين يحتاجون إلى أساس برمجي منظم يدعم التقدم نحو تطبيقات أكثر تعقيداً.

- المبتدئون في تطوير البرمجيات.
- خريجو التخصصات التقنية الراغبون في تعلم Java.
- موظفو تقنية المعلومات المنتقلون إلى البرمجة.
- مطورو التطبيقات الجدد في فرق البرمجيات.
- المشاركون الراغبون في بناء أساس قبل Spring أو Android.

### اليوم الأول: مدخل إلى Java وبيئة التطوير

- خصائص Java واستخداماتها في تطوير التطبيقات.
- مكونات JDK وJRE وJVM ودورها العملي.
- إعداد بيئة التطوير وتشغيل أول برنامج.
- هيكل ملف Java وطريقة تنفيذ البرنامج.
- استخدام محرر الكود وسطر الأوامر بفاعلية.

### اليوم الثاني: أساسيات الصياغة البرمجية

- القواعد الأساسية لكتابة التعليمات البرمجية.
- أنواع البيانات البدائية والمراجع.
- المتغيرات والثوابت ونطاق التعريف.
- العمليات الحسابية والمنطقية والمقارنة.
- تنسيق المخرجات وقراءة المدخلات البسيطة.

### اليوم الثالث: التحكم في مسار التنفيذ

- استخدام جمل if وelse لاتخاذ القرار.
- تطبيق switch في الحالات متعددة الفروع.
- بناء شروط واضحة قابلة للاختبار.
- تجنب الأخطاء المنطقية في فروع التنفيذ.
- تحليل سيناريوهات عملية لاتخاذ القرار.

### اليوم الرابع: الحلقات والتكرار

- استخدام for في التكرار المحدد.
- استخدام while وdo while في التكرار الشرطي.
- التحكم في التكرار باستخدام break وcontinue.

- معالجة حالات التكرار غير المنتهي.
- تطبيقات عملية على العد والبحث البسيط.

### اليوم الخامس: المصفوفات وهياكل البيانات الأساسية

- تعريف المصفوفات أحادية ومتعددة الأبعاد.
- قراءة عناصر المصفوفة وتحديثها.
- البحث والفرز البسيط داخل المصفوفات.
- استخدام ArrayList لإدارة مجموعات مرنة.
- اختيار الهيكل المناسب للمشكلة البرمجية.

### اليوم السادس: الدوال وتنظيم الكود

- تعريف الدوال واستدعاؤها داخل البرنامج.
- تمرير المعاملات وإرجاع القيم.
- تجزئة المشكلات إلى وحدات قابلة للإدارة.
- استخدام التحميل الزائد للدوال بشكل مناسب.
- تحسين قابلية القراءة وإعادة الاستخدام.

### اليوم السابع: مبادئ البرمجة الكائنية

- مفهوم الكائن والفئة في Java.
- تعريف الخصائص والسلوكيات داخل الفئات.
- إنشاء الكائنات واستخدامها في البرامج.
- التمييز بين الحالة والسلوك والمسؤولية.
- تطبيق نموذج بسيط قائم على الكائنات.

### اليوم الثامن: التغليف والمنشآت

- استخدام محددات الوصول لتنظيم الصلاحيات.
- تطبيق التغليف لحماية حالة الكائن.
- تعريف المنشآت وتهيئة القيم الأولية.

- استخدام setters و getters بطريقة سليمة.
- تقليل الاعتمادية المباشرة بين المكونات.

### اليوم التاسع: الوراثة وتعدد الأشكال

- تطبيق الوراثة لإعادة استخدام السلوك.
- استخدام super للوصول إلى عناصر الأصل.
- إعادة تعريف الدوال وفق الحاجة.
- فهم تعدد الأشكال في وقت التشغيل.
- تحديد حدود استخدام الوراثة عملياً.

### اليوم العاشر: الواجهات والفئات المجردة

- تمييز الواجهات عن الفئات المجردة.
- تصميم عقود برمجية باستخدام interfaces.
- استخدام abstract classes لمشاركة السلوك.
- تطبيق مبادئ الفصل بين التعريف والتنفيذ.
- اختيار الأسلوب الأنسب للتصميم البسيط.

### اليوم الحادي عشر: معالجة الأخطاء والاستثناءات

- فهم أنواع الأخطاء في Java.
- استخدام try و catch لمعالجة الاستثناءات.
- تنفيذ finally لإغلاق الموارد.
- إنشاء استثناءات مخصصة للحالات الخاصة.
- كتابة رسائل أخطاء واضحة ومفيدة.

### اليوم الثاني عشر: التعامل مع النصوص والتواريخ

- استخدام String و StringBuilder بفاعلية.
- تطبيق عمليات البحث والقص والاستبدال.
- تنسيق النصوص للعرض والمعالجة.

- استخدام LocalDate و LocalDateTime.
- معالجة التواريخ والفترات الزمنية البسيطة.

### اليوم الثالث عشر: التعامل مع الملفات والمدخلات

- قراءة الملفات النصية باستخدام Java APIs.
- كتابة البيانات إلى الملفات بأمان.
- إدارة الموارد باستخدام try-with-resources.
- معالجة أخطاء الملفات والمسارات.
- تنظيم البيانات النصية للمعالجة اللاحقة.

### اليوم الرابع عشر: المجموعات Collections

- استخدام List و Set و Map في التطبيقات.
- اختيار المجموعة المناسبة لكل حالة.
- التكرار على المجموعات بطرق مختلفة.
- فهم المساواة والترتيب داخل المجموعات.
- تطبيقات عملية على إدارة بيانات بسيطة.

### اليوم الخامس عشر: البرمجة الوظيفية الأساسية

- استخدام lambda expressions في الحالات البسيطة.
- فهم functional interfaces ودورها.
- تطبيق Streams لمعالجة المجموعات.
- تنفيذ filter و map و collect.
- تحسين وضوح الكود عند معالجة البيانات.

### اليوم السادس عشر: مبادئ التصميم والكود النظيف

- تسمية المتغيرات والدوال بوضوح.
- تقليل التكرار وتحسين تنظيم الفئات.
- تطبيق مبادئ المسؤولية الواحدة.

- فصل منطق الأعمال عن منطق العرض.
- مراجعة كود بسيط وتحسين بنيته.

### اليوم السابع عشر: الاختبار وتصحيح الأخطاء

- استخدام أدوات التصحيح داخل بيئة التطوير.
- تتبع القيم أثناء تنفيذ البرنامج.
- كتابة اختبارات وحدة أساسية.
- فهم حالات الاختبار والنتائج المتوقعة.
- تحليل الأخطاء الشائعة وتصحيحها منهجياً.

### اليوم الثامن عشر: أساسيات قواعد البيانات مع Java

- فهم دور JDBC في ربط التطبيقات.
- إنشاء اتصال مبسط بقاعدة بيانات.
- تنفيذ أوامر قراءة وتحديث أساسية.
- معالجة نتائج الاستعلامات بطريقة منظمة.
- تطبيق مبادئ السلامة في الاستعلامات.

### اليوم التاسع عشر: بناء تطبيق Java مصغر

- تحديد متطلبات تطبيق تدريبي بسيط.
- تصميم الفئات والعمليات الرئيسية.
- تنفيذ منطق التطبيق خطوة بخطوة.
- اختبار الوظائف الأساسية وتصحيح الأخطاء.
- تحسين العرض والتنظيم النهائي للكود.

### اليوم العشرون: المراجعة والتطبيق الختامي

- مراجعة المفاهيم الأساسية للبرمجة بلغة Java.
- حل تمارين تطبيقية متعددة المستويات.
- مناقشة أفضل ممارسات التعلم المستمر.

- تقديم المشروع المصغر ومراجعة نتائجه.
- بناء خطة تطوير مهارية بعد البرنامج.

## course\_daily\_schedule

يمتد البرنامج لمدة 20 يوماً تدريبياً، بواقع 4 ساعات تدريبية يومياً. يبدأ كل يوم بمراجعة قصيرة للمفاهيم السابقة، ثم شرح للمحور الرئيسي، يتبعه تطبيق عملي مباشر على الكود، وينتهي اليوم بتمرين أو مراجعة موجزة لتعزيز الفهم التدريجي.

## course\_assessment

يعتمد التقييم على المشاركة اليومية، والتمارين العملية، ومراجعة الكود، وتنفيذ مشروع تدريبي مصغر في نهاية البرنامج. يحصل المشاركون على شهادة حضور/إتمام من AINFCT وفق متطلبات الحضور والمشاركة المعتمدة.

## course\_key\_competencies

- أساسيات Java.
- البرمجة الكائنية.
- معالجة الأخطاء.
- المجموعات والملفات.
- تصميم كود منظم.
- الاختبار والتصحيح.

## مكتب القاهرة الرئيسي

القاهرة، جمهورية مصر العربية  
7 شارع وهران، الطيران، مدينة نصر

201152466358+

info@ainfct.com

ainfct.com

رقم التسجيل الضريبي: 472920235

## مكتب مدريد الفرعي

مدريد، إسبانيا

شارع الصحة 3، وسط المدينة، 28013 مدريد

training@ainfct.com

ainfct.com