Pearson
BTEC

شهادة Pearson BTEC International
المستوى 3

تكنولوجيا المعلومات

الوحدة 7

تطوير تطبيقات الأجهزة المحمولة

المؤلفون: مارك فيشبول وبيرني فيشبول

المستوى 3

كتاب الطالب

نُشِرَ بواسطة شركة بيرسون إديوكيشن ليمتد، 80 ستراند، لندن، WC2R 0RL.

www.pearsonschoolsandfecolleges.co.uk

يمكن العثور على نسخ من المواصفات الرسمية لجميع شهادات Pearson على الموقع الإلكتروني:
qualifications.pearson.com

© حقوق التأليف والنشر لعام 2023 محفوظة لصالح شركة بيرسون إديوكيشن ليمتد

حُرِّرَ بواسطة شركة فلورنس برودكشن ليمتد

طُبع بواسطة شركة فلورنس برودكشن ليمتد

© حقوق التأليف والنشر للرسوم التوضيحية الأصلية محفوظة لشركة بيرسون إديوكيشن ليمتد

الرسوم التوضيحية من إعداد شركة فلورنس برودكشن ليمتد

نُشرت هذه الطبعة عام 2024

ما لم يُذكر خلاف ذلك هنا، فإن أي علامات تجارية لطرف ثالث قد تظهر في هذا العمل هي ملك لأصحابها المعنيين وأي إشارات إلى العلامات التجارية أو الشعارات أو الشكل التجاري الآخر لطرف ثالث هي لأغراض توضيحية أو وصفية فقط. لا يُقصد من هذه الإشارات أي رعاية أو تأييد أو تفويض أو ترويج لمنتجات بيرسون إديوكيشن ليمتد من قبل مالكي هذه العلامات، أو أي علاقة بين المالك وشركة بيرسون إديوكيشن ليمتد أو الشركات أو المؤلفين أو المرخص لهم أو الموزعين التابعين لها.

فهرسة المكتبة البريطانية في بيانات النشر

يتوافر سجل كتالوج لهذا الكتاب من المكتبة البريطانية

إشعار حقوق التأليف والنشر

جميع الحقوق محفوظة. لا يجوز إعادة إنتاج أي جزء من هذا المنشور بأي شكل أو بأي وسيلة (بما في ذلك نسخه أو تخزينه في أي وسيط باستخدام الوسائل الإلكترونية، سواء بشكل عابر أو عرضي لبعض الاستخدامات الأخرى لهذا المنشور) دون إذن كتابي من مالك حقوق التأليف والنشر، باستثناء الحالات التي تتوافق مع أحكام قانون حقوق التأليف والنشر والتصاميم وبراءات الاختراع لعام 1988 أو بموجب شروط ترخيص صادر عن وكالة ترخيص حقوق التأليف والنشر، برناردز إن، 86 فيتر لين، لندن EC4A 1EN (www.cla.co.uk). يجب توجيه طلبات الحصول على إذن كتابي لمالك حقوق التأليف والنشر إلى الناشر.

المواقع الإلكترونية

لا تتحمل بيرسون إديوكيشن ليمتد المسؤولية عن المحتوى الخاص بأي مواقع إنترنت خارجية. ومن الضروري أن يعاين المعلمون كل موقع إلكتروني قبل استخدامه في الصف للتأكد من أن عنوان URL ما يزال دقيقًا، وملائمًا، ومناسبًا. ونقترح أن يقوم المعلمون بوضع إشارة مرجعية على المواقع الإلكترونية المفيدة والنظر في تمكين المتعلمين من الوصول إليها من خلال الشبكة الداخلية للمدرسة/ الكلية.

ملاحظة من الناشر

تنفذ Pearson عمليات تحرير مُحكمة لضمان دقة المحتوى في هذا المنشور، ويتم بذل كل جهد ممكن لضمان خلو هذا المنشور من الأخطاء. ولكن، ما نحن إلا بشر، وأحيانًا تحدث أخطاء. ولا تتحمل Pearson المسؤولية عن أي سوء فهم ينشأ نتيجة أخطاء في هذا المنشور، ولكن من أولوياتنا ضمان دقة المحتوى. إذا لاحظت وجود خطأ، فيُرجى التواصل معنا عبر resourcescorrections@pearson.com حتى نتأكد من تصحيحه.

في حين بذل الناشرون قصارى جهدهم لضمان دقة المشورة بشأن التأهيل وتقييمه، فإن المواصفات الرسمية ومواد دليل التقييم المرتبطة بها تمثل المصادر الموثوقة الوحيدة للمعلومات وينبغي الرجوع إليها دائمًا للحصول على إرشادات نهائية.

شكر وتقدير

الصور:

123آر اف: ص 12 (الشكل 7.8)؛ **ألامي ستوك فكتور**: إيان دانغال ص 6 (الشكل 7.3)؛ **أوداسيتي**: لقطة شاشة لأوداسيتي © 2024 بواسطة أوداسيتي ص16 (الشكل 7.13)؛ **غاما بلاي**: لقطة شاشة لتطبيق هاتف محمول، تُستخدم بإذن من غاما بلاي ص 5 (الشكل 7.1)؛ شركة **جوجل المحدودة**: لقطات شاشة لنظام Android Studio p.9 (الشكل 7.6)، ص 19 (الشكل 7.18)، ص 20 (الشكل 7.19 و7.20)، ص 31 (الشكل 7.33 و7.34)، ص 32 (الشكل 7.35 و7.36)، ص 33 (الشكل 7.38 و7.39)؛ شترستوك: أليكسي بولدين ص 6 على اليمين (الشكل7.2)، نيكي ص 6 على اليسار (الشكل 7.2)، أندراي بوبوف ص32 (الشكل 7.37)، مايكل جونج ص 41.

© جميع حقوق طبع ونشر الصور الأخرى محفوظة لصالح شركة بيرسون إديوكيشن

تعرُّف الوحدة

الأجهزة المحمولة موجودة في كل مكان، ومبيعاتها المذهلة مدفوعة بشكل أساسي بالتطبيقات المبتكرة التي تحافظ على تعليم مستخدميها وإطلاعهم على المعلومات وكذلك الترفيه. يتمتع قطاع تطبيقات الأجهزة المحمولة برواج تجاري مزدهر ويمكن استهدافه بشكل مربح من قبل الشركات الناشئة الصغيرة والشركات الكبيرة على حد سواء.

وعادةً ما يكون تطوير تطبيق جهاز محمول نتيجة وجود فكرة أولية جيدة أو حل مشكلة أساسية؛ ويأتي نجاح تطبيق الجهاز المحمول هذا من تقدير تعقيدات الأجهزة المحمولة والأشكال المختلفة للوظائف المتاحة وكيفية تصميم التطبيقات لتكون قابلة للاستخدام بدون حاجة للتعلم أو التدريب. تعد القدرة على إنشاء مثل هذه التطبيقات مهارة مطلوبة للغاية وستساعدك، كمطور برامج، على اكتساب ميزة تنافسية.

كيفية إجراء التقييم

سيتم تقييم هذه الوحدة داخليًا من خلال مهمتين يحددهما معلمك. ستجد خلال هذه الوحدة أنشطة تمرين تقييمي ستساعدك على العمل حتى تصل إلى تقييمك. وإنجاز هذه الأنشطة لا يعني أنك قد حصلت على درجة معينة، ولكنك ستكون قد نفذت بحثًا أو تدريبًا مفيدًا سيفيدك عندما يتعلق الأمر بواجبك النهائي.

لكي تتمكن من إكمال المهام في واجباتك بنجاح، من المهم التحقق من أنك استوفيت جميع معايير التقدير اللازم للنجاح. ويمكنك القيام بذلك خلال عمل الواجب المُحدد.

إذا كنت تهدف إلى تحقيق درجة التفوق أو الامتياز، يجب عليك أيضًا التأكد من تقديم المعلومات في مهمتك بالأسلوب الذي تتطلبه معايير التقييم ذات الصلة. على سبيل المثال، تتطلب معايير التفوق منك المراجعة والتبرير بينما تتطلب معايير الامتياز منك التقييم.

ستتألف الواجبات التي حددها معلمك من سلسلة من المهام المصممة لتلبية المعايير الواردة في الجدول. من المرجح أن تتكون المهمة 1 بشكل أساسي من مهمة مكتوبة قائمة على البحث تتطلب منك دراسة تطبيقات الأجهزة المحمولة وأنواع الأجهزة المحمولة، بينما ستشمل المهمة 2 أنشطة عملية، مثل:

- تصميم تطبيق جهاز محمول يستخدم وظائف الجهاز
- تطوير تطبيق جهاز محمول يستخدم وظائف الجهاز.

التقييم

سيُجرى التقييم من خلال سلسلة من المهام التي يحددها معلمك.

معايير التقييم

يوضح لك هذا الجدول ما يجب عليك القيام به من أجل الحصول على درجة النجاح أو التفوق أو الامتياز.

النجاح	التفوق	الامتياز
نتاج التعلم أ دراسة تطبيقات الأجهزة المحمولة وأنواع الأجهزة المحمولة		
A.P1 شرح كيف يؤثر الغرض من تطبيق الجهاز المحمول واحتياجات المستخدم وتفضيلاته وخصائصه في تصميم التطبيق والميزات المقدمة فيه. **تمرين تقييمي 7.1** **A.P2** شرح تأثير التقنيات الحالية في تصميم تطبيقات الأجهزة المحمولة وتنفيذها. **تمرين تقييمي 7.1**	**A.M1** تحليل كيفية تأثر تنفيذ وتصميم تطبيقات الأجهزة المحمولة بالمستخدم المقصود والتقنيات الحالية والغرض من التطبيق. **تمرين تقييمي 7.1**	**A.D1** تقييم مدى تأثر فعالية تنفيذ وتصميم تطبيقات الأجهزة المحمولة بالمستخدم المقصود والتقنيات الحالية والغرض من التطبيق. **تمرين تقييمي 7.1**
نتاج التعلم ب تصميم تطبيق جهاز محمول يستخدم وظائف الجهاز		
B.P3 إعداد تصميمات لتطبيق جهاز محمول لتلبية المتطلبات المحددة. **تمرين تقييمي 7.2** **B.P4** مراجعة تصميمات تطبيقات الأجهزة المحمولة مع الآخرين لتحديد التحسينات وتطبيقها. **تمرين تقييمي 7.2**	**B.M2** تبرير كيف تضمن القرارات التي اتُّخذت خلال عملية التصميم أن التصميم الخاص بالتطبيق سيلبي المتطلبات المحددة. **تمرين تقييمي 7.2**	**BC.D2** تقييم التصميم وتطبيق الجهاز المحمول المحسّن وفقًا لمتطلبات العميل. **تمرين تقييمي 7.2**
نتاج التعلم ج تطوير تطبيق جهاز محمول يستخدم وظائف الجهاز		
C.P5 إنتاج تطبيق جهاز محمول يلبي معايير التصميم. **تمرين تقييمي 7.2** **C.P6** اختبار تطبيق جهاز محمول من حيث كفاءة العمل وسهولة الاستخدام والاستقرار والأداء. **تمرين تقييمي 7.2** **C.P7** مراجعة مدى تلبية تطبيق الجهاز المحمول للمتطلبات المحددة. **تمرين تقييمي 7.2**	**C.M3** تحسين تطبيق جهاز محمول يُلبي معايير التصميم. **تمرين تقييمي 7.2**	**BC.D3** إظهار المسؤولية الفردية والإبداع والإدارة الذاتية الفعالة في تصميم وتطوير ومراجعة تطبيق الجهاز المحمول. **تمرين تقييمي 7.2**

بدء النشاط

يتطلب تطوير تطبيقات الأجهزة المحمولة معرفة وتطبيق العديد من المهارات المختلفة. اكتب قائمة بالأدوات والتقنيات التي تعتقد أنك ستحتاجها لإنشاء تطبيقات الأجهزة المحمولة المبتكرة. عند الانتهاء من هذه الوحدة، تحقق مما إذا كانت قد فاتتك أي مهارات أو أدوات أو تقنيات واضحة من هذه القائمة.

نتاجات التعلم

ستتعلم في هذه الوحدة:

أ} دراسة تطبيقات الأجهزة المحمولة وأنواع الأجهزة المحمولة

ب} تصميم تطبيق جهاز محمول يستخدم وظائف الجهاز

ج} تطوير تطبيق جهاز محمول يستخدم وظائف الجهاز

أ} دراسة تطبيقات الأجهزة المحمولة وأنواع الأجهزة المحمولة

قبل البدء في تطوير تطبيقات الأجهزة المحمولة، من المهم الحصول على تقدير لنطاق الأجهزة المعنية وكفاءة عملها. يتم تشكيل العديد من قرارات التصميم من خلال الحريات والقيود المفروضة على الأجهزة المستهدفة. في هذا الجزء، سنبحث في الأنواع المختلفة من تطبيقات الأجهزة المحمولة والأجهزة التي تعمل عليها والميزات والخصائص والخيارات التي ستشكل مسار التطوير الخاص بك.

المهارات

- مهارات التحليل واتخاذ القرار
- التواصل الكتابي الرسمي

أنواع تطبيقات الأجهزة المحمولة

تطبيقات الأجهزة المحمولة (المعروفة أكثر باسم تطبيقات الهاتف) هي برامج مصممة للعمل على الهواتف الذكية وأجهزة الحاسوب اللوحية والتكنولوجيا الناشئة القابلة للارتداء مثل الساعات والنظارات. عادةً ما يتم تطوير التطبيق كواحد من ثلاثة أنواع عامة موضحة في الجدول 7.1.

تتطلب التطبيقات الأصلية والهجينة برامج متخصصة (وأحيانًا أجهزة) ليتم تطويرها بنجاح. على سبيل المثال، يتم إنشاء تطبيقات Apple® iPad وiPhone عادةً باستخدام برامج وأجهزة خاصة بنظام Apple تضمن معايير وممارسات التطوير.

في المقابل، يمكن تطوير تطبيقات الويب باستخدام أدوات تحرير النصوص البسيطة مثل Microsoft® Notepad وقد تعمل بشكل جيد بالقدر نفسه من داخل برنامج عميل متصفح الويب الذي يعمل على حاسوب مكتبي أو وحدة التحكم في الألعاب أو جهاز محمول. ونظرًا إلى تعدد استخداماتها، فإنها ذات جدوى استثمارية أكبر (من حيث الوقت والمال وما إلى ذلك) من قبل المطورين لأن قاعدة المستخدمين المحتملين أكبر بكثير.

بشكل عام، عند اختيار نوع تطبيق الجهاز المحمول المراد تطويره، هناك مفاضلة بين زمن التطوير والتكلفة والوظائف المطلوبة ومستوى تجربة المستخدم المطلوبة.

الجدول 7.1 الأنواع الثلاثة لتطبيقات الأجهزة المحمولة

التطبيقات الأصلية (Native apps)	التطبيقات المستندة إلى الويب (Web apps)	التطبيقات الهجينة (Hybrid apps)
• تم تصميمها وتطويرها لنظام تشغيل معين. • يمكن الاستفادة من جميع الوظائف (المسموح بها) المتوافرة في الجهاز، مثل الموقع والكاميرا وجهات الاتصال. • عادةً ما تحاكي التطبيقات (الرسمية) الخاصة بالشركة المصنعة، بالاعتماد على تجربة المستخدم السابقة لسهولة الاستخدام.	• التطبيقات البعيدة، التي تعمل عادةً على الخادم. • يتفاعل المستخدم عادةً مع التطبيق من خلال متصفح الويب الخاص بجهاز محمول. • لا يتم عادةً تثبيت أي تطبيق أو بيانات على الجهاز. • تقييم وظائف الجهاز محدود. • قد تحاكي بصريًا الواجهة المعتادة للجهاز.	• عادة ما تكون متوافقة مع منصات متعددة. • يستخدم عادةً مكونات الموقع الإلكتروني التي يتم (تغليفها) داخل تطبيق أصلي. • قد يكون الوصول إلى وظائف الجهاز محدودًا. • قد يُنظر إليه على أنه نهج تطوير أكثر فعالية من حيث التكلفة، خاصة عندما يتم استهداف أنواع متعددة من الأجهزة المحمولة.

سياق تطبيقات الأجهزة المحمولة

يتم إنشاء تطبيقات الأجهزة المحمولة لتلبية العديد من احتياجات المستخدم المختلفة، وهذا يحدد سياقها والغرض منها. وتم تصميم كل تطبيق لتزويد المستخدم بأداة أو تجربة أو تحسين معين لا يوفره جهازه المحمول الأساسي (أو ليعمل بشكل جيد). يعتمد التصميم العام للواجهة وتحديات التطوير والاستخدام الفعلي بمجرد التثبيت على الجهاز المحمول على نوع التطبيق.

مناقشة

إذا كان لديك جهاز محمول، فمن المحتمل أنك قابلت مجموعة متنوعة من تطبيقات الأجهزة المحمولة المختلفة. فكر في هذه التطبيقات، وخاصة مظهرها ووظائفها، وحدد ما إذا كان من المحتمل أن تكون تطبيقات أصلية أو ويب أو هجينة. ناقش استنتاجاتك مع صفك.

الجدول 7.2 الفئات الشائعة لسياق التطبيق والغرض منه

السياق	الغرض والميزات	أمثلة
الموقع	تطبيقات مصممة لتزويدك بالمعلومات الجغرافية بناءً على موقعها الحالي (مثل الخرائط ومكتشفات مسارات GPS وتجارب الواقع المعزز (AR)).	Google™ Maps, Wikitude, Yelp Monocle, Google Sky, Aurasma
الأداة	التطبيقات المقدمة للمساعدة في تكوين جهازك أو صيانته (مثل مدير الملفات (يضيف الملفات أو يزيلها)، وأدوات النسخ الاحتياطي (النسخ الاحتياطي إلى السحابة)، وشاشات النظام).	Glary Utilities, CCleaner, WinZip, AVG AntiVirus for Android
المجموعات الإنتاجية	تطبيقات توفر وظائف على غرار المكتب لمعالجة الكلمات وجداول البيانات وعروض الشرائح وقواعد البيانات. إنها توفر للمستخدم تسهيلات لكتابة الرسائل أو التقارير، وإنشاء العروض التقديمية، وحساب تكاليف المشروع، وما إلى ذلك.	Adobe® Acrobat; Microsoft Office, Office 365 and OneNote; Dropbox
شاشة كاملة غامرة	تعرض التطبيقات التي تستخدم الشاشة بالكامل لجذبك تمامًا (مثل الألعاب).	Temple Run 2, Candy Crush Saga, Angry Birds, Fruit Ninja, Minecraft: Pocket Edition
أسلوب الحياة	تطبيقات مصممة لإثراء حياتك (مثل الوصفات الصحية، والأعمال اليدوية، والتصميم الداخلي، والبستنة، والسفر). قد يقدمون تعليمات مرئية خطوة بخطوة وسردية ونماذج ثلاثية الأبعاد وروابط للمواقع الإلكترونية التجارية.	Craftsy, Gumtree, TripAdvisor, Etsy, Evernote, Google Translate, eBay
الاجتماعية	تطبيقات مصممة لتوصيلك بأشخاص آخرين، للمساعدة في التواصل ومشاركة الأفكار والأحداث والصور ومقاطع الفيديو.	X (المعروف سابقًا باسم Twitter)، Facebook, Snapchat, WhatsApp, Pinterest, Instagram, WordPress, Blogger
الترفيه	التطبيقات التي توفر محتوى وسائط (مثل مشغلات الموسيقى وبث الفيديو والبودكاست وقارئات الكتب الإلكترونية).	YouTube, BBC iPlayer, Spotify, Shazam, SoundCloud, Kindle, Audible
التطبيقات المصغرة	التطبيقات التي تستهلك القليل جدًا من شاشة الجهاز المحمول والتي توفر وصولًا سريعًا إلى البيانات أو الإعدادات المباشرة (مثل مؤشرات الأخبار وإعدادات الجهاز السريعة ومرافق البحث والتقويم والمواعيد).	Feedly, Flashlight apps, Todoist, BBC News

يشكل إنشاء تطبيقات لكل فئة من هذه الفئات تحديات مختلفة لمطور تطبيقات الأجهزة المحمولة.

بعبارات بسيطة، سيتم تحديد تطبيق الجهاز المحمول الذي تصممه وميزاته من خلال المهمة (المهام) التي يجب أن يؤديها والتفضيلات والاحتياجات الشخصية للمستخدم المستهدف.

قد يكون تحديد احتياجات المستخدم المحددة أمرًا صعبًا، ولكن هناك بعض النقاط والأسئلة الرئيسة التي يجب تذكرها:

- احتياجات المستخدم - ما الوظيفة التي يحتاجها المستخدم من التطبيق؟ ما الميزات المطلوبة؟
- تفضيلات المستخدم - كيف يريد المستخدم استخدام التطبيق؟ كيف يريد المستخدم التفاعل مع التطبيق؟ كيف يريد المستخدم التنقل بين وظائف التطبيق وميزاته؟ ما النمط المرئي الذي يجب أن يقدمه التطبيق؟
- خصائص المستخدم - كيف يجب أن يلبي التطبيق أنواعًا مختلفة من المستخدمين، مثل عمر المستخدم، والخبرة الفنية للمستخدم، والإعاقات مثل ضعف البصر أو صعوبات السمع، والبيئة المادية التي يستخدم فيها التطبيق، ومدى إمكانية تخصيص إعدادات التطبيق لتناسب المستخدم، ومستوى الدعم والمساعدة التي يقدمها التطبيق؟

تؤدي الإجابة عن هذه الأنواع من الأسئلة إلى إنشاء ملف تعريف للمستخدمين المستهدفين للتطبيق وتساعد في تشكيل كل من تصميمه والميزات المضمنة.

ربما قد واجهت موقفًا قمت فيه بتحميل تطبيقين متشابهين على هاتفك يدعيان تقديم الوظيفة نفسها؛ تصميم كل منهما وفعاليته وسهولة استخدامه ستمكنك من تحديد أيهما يستحق الاحتفاظ وأيهما يجب حذفه

تطبيق النظرية

أصبح البث الصوتي شكلًا شائعًا من أشكال الترفيه في السنوات القليلة الماضية وهناك العديد من تطبيقات البودكاست (برامج Podcast هي برامج تعمل على تشغيل الملفات الصوتية بميزات تنوع المحتوى ومصنفة بطريقة إحترافية، وأهم مزاياها هي تحميل الملفات الصوتية والاستماع إليها دون الحاجة للاتصال بالإنترنت) المختلفة المتاحة لمنصات الهواتف المحمولة الشهيرة.

حدد ثلاثة تطبيقات بودكاست مختلفة لمنصة محمولة محددة وتحقق من تصميمها ووظائفها وميزاتها وأدائها وسهولة استخدامها.

أي واحد توصي به لصديق؟ اشرح وبرر أسبابك

تكامل الأجهزة المحمولة

تأتي الأجهزة المحمولة بأشكال وأحجام مختلفة (الهواتف الذكية والأجهزة اللوحية والأجهزة القابلة للارتداء). تختلف ميزاتها ووظائفها أيضًا بشكل كبير. ويرجع ذلك جزئيًا إلى حقيقة أن الشركات المصنعة للهواتف المحمولة ترغب في تمييز منتجاتها في السوق ويمكن أن يكون دمج أجهزة استشعار محددة (أمان بصمات الأصابع، على سبيل المثال) نقطة بيع جذابة. عند تصميم تطبيق هاتف محمول، من الضروري أن تكون على دراية بهذه الخصائص المختلفة والآثار التي قد تكون لها على كل من التصميم والتطوير.

وظائف الجهاز المحمول

بصفتك مطورًا، لا يمكنك ضمان أن الجهاز المحمول له وظيفة معينة عند تصميم التطبيق وتطويره، لذلك يوصى بالتحقق من نظام التشغيل لمعرفة ما إذا كانت الوظيفة متاحة. ومع ذلك، يمكن ضمان الوظائف الرئيسة مثل الاهتزاز ومخرج سماعة الرأس ومكبر الصوت وشاشات اللمس والميكروفون وكاميرات الصور والفيديو بشكل عام عبر أي جهاز محمول حديث، بغض النظر عن الشركة المصنعة.

الجدول 7.3 وحدات الاستشعار

وظيفة الجهاز المحمول	ما هي	كيفية الاستخدام
مقياس التسارع	جهاز استشعار الحركة الذي يقيس القوة الواقعة على الجهاز على جميع المحاور الثلاثة (x و y و z)، *باستثناء الجاذبية*.	تستخدم لتتبع الحركة مثل هز الجهاز وتتبع الفرق بين المشي أو الركض.
مقياس المغناطيسية	جهاز استشعار موضعي يقيس المجال المغناطيسي المحيط.	يستخدم لإنشاء بوصلة.
ترمومتر	يُسمى أيضًا ثيرموميتر وهو مستشعر الحرارة الذي يقيس درجة الحرارة المحيطة أو الداخلية، عادةً بالدرجات المئوية (C °).	ويستخدم لقياس درجة حرارة الغرفة أو درجة حرارة الجهاز المحمول نفسه.
مقياس الضغط الجوي	جهاز استشعار محيطي يقيس ضغط الهواء عادة إما بالهيكتوباسكال (hPa) وإما المليبار (mbar).	يستخدم لقياس ضغط الهواء في الغرفة. ويستخدم أيضًا لقياس الارتفاع، لجعل GPS أكثر دقة في الأماكن المرتفعة.
فوتومتر (مستشعر الإضاءة)	يطلق عليه اسم آخر أيضًا بالإنجليزية (ambient light)، وهو مستشعر شدة الضوء مثل التحكم بسطوح الشاشة، وتبرز أهم استخداماته عند التصوير لضبط جودة الكاميرا عند التصوير، ويقيس مستويات الإضاءة المحيطة باستخدام وحدة لوكس (lx).	يُستخدم لضبط سطوع الشاشة تلقائيًا وفقًا لظروف الإضاءة الخارجية لتحسين إمكانية القراءة والحفاظ على الطاقة.
الاتجاه	جهاز استشعار الحركة الذي يحسب الميل والاتجاه (انظر الشكل 7.1). يستخدم مقياس التسارع.	يُستخدم لتغيير اتجاه شاشة التطبيق عندما يتم تغيير الجهاز من الوضع الرأسي إلى الوضع الأفقي، أو لإنشاء مستوى الاستواء.
نظام تحديد المواقع العالمي (GPS)	جهاز استشعار موضعي يستخدم الأقمار الصناعية أو المحطات الخلوية لتحديد الموقع الفعلي.	يُستخدم لتتبع الموقع الفعلي للجهاز المحمول لتوفير معلومات حساسة عن الموقع (مثل أقرب مطعم أو محطة وقود) أو الخدمات (العثور على أصدقاء) والخرائط الحية وتعرف المسار.

الشكل 7.1 تطبيق ميزان التسوية على هاتف محمول يستخدم مستشعر التوجيه

واجهة المستخدم

واجهة المستخدم (غالبًا ما يتم اختصارها إلى UI)، هي مزيج من البرامج والأجهزة التي يتفاعل معها المستخدم لأداء مهام المحددة. يتضمن جزء البرنامج من واجهة المستخدم نظام التشغيل الأساسي وتطبيق الجهاز المحمول المحدد نفسه. ويتم تمثيل أجهزة واجهة المستخدم من خلال أجهزة الإدخال (مثل شاشات اللمس والأزرار المادية والكاميرا والميكروفون) وأجهزة الإخراج (مثل الشاشة ومكبر الصوت والاهتزاز).

يعد تطوير تطبيقات الأجهزة المحمولة أمرًا صعبًا نظرًا لوجود العديد من القيود التي يحتاجها المصمم والمبرمج للتنقل بنجاح لإنشاء تجربة تفاعلية فعالة لواجهة المستخدم. تشمل هذه الخصائص الصعبة:

- حجم شاشة محدود (أو متغير)
- آلية إدخال لوحة المفاتيح/لوحة مفاتيح محدودة
- قوة معالجة محدودة (على الرغم من تحسن هذا بسرعة في أجهزة الهواتف الذكية الحديثة).

يتطلب استخدام آليات الإدخال البديلة مثل الصوت والتحكم باللمس والإيماءات المعقدة (الضغط والتمدد والتمرير) والتأثيرات المادية (الاهتزاز والإمالة) تفكيرًا مكثفًا قبل استخدامها في التطبيق، من أجل تجربة سهلة الاستخدام. وعند استخدامها بشكل جيد، على سبيل المثال في تطبيق رسم يسمح للمستخدم بهز الجهاز لمسح الصورة، تكون النتائج مرضية والتجربة بديهية للغاية. في التصميم السيئ، على سبيل المثال التحكم في شخصية ذكية قابلة للعب باستخدام عناصر تحكم في شاشة لمس منخفضة الكفاءة، يمكن أن تكون النتائج غير دقيقة ومحبطة للمستخدمين.

المصطلحات الرئيسة

شاشات اللمس المقاومة - تستخدم طبقتين (عادة الزجاج والبلاستيك) مغطى بمواد موصلة للكهرباء (عادة أكسيد الإنديوم القصديري (ITO)). يتم إبقاء الطبقتين منفصلتين حتى يضغط إصبع أو قلم على بعضهما البعض، مما يتسبب في تغيير موضعي في المقاومة الكهربائية. هذا النوع من الشاشات اللمسية أقل تكلفة في التصنيع من الشاشات اللمسية السعوية ولكنه ليس حساسًا للغاية ولا يمكنه دعم إيماءات اللمس المتعدد.

شاشات اللمس السعوية - تستخدم طبقتين متباعدتين من الزجاج المطلي بمكثفات أكسيد القصدير والإنديوم (ITO) الدقيقة. عندما يلمس إصبع المستخدم الشاشة، فإنه يغير المجال الكهروستاتيكي المحلي للشاشة. هذا النوع من شاشات اللمس أكثر سطوعًا وحساسية من شاشات اللمس المقاومة ويدعم إيماءات اللمس المتعدد. ومع ذلك، فإن التعقيد الهيكلي للشاشة يرفع تكاليف الإنتاج.

حتى الأجهزة من نفس العائلة (على سبيل المثال Apple iPhone أو iPad) قد تحتوي على خيارات واجهة مستخدم مختلفة قليلاً، لذلك يجب ألا تأخذ أي ميزة كأمر مسلم به دون البحث عنها بشكل صحيح.

قد تتضمن الميزات المتغيرة نوع شاشة اللمس (**المقاومة** أو **السعوية**، ذات لمس واحد أو متعدد)، أزرارًا مادية مخصصة أو مجرد شاشة صغيرة جدًا. يحتاج مطورو تطبيقات الأجهزة المحمولة إلى التفكير في كيفية عمل التطبيقات ضمن قيود واجهات المستخدم المختلفة (وعلى الأجهزة المختلفة) لدعم المستخدمين المستهدفين بشكل فعّال.

نظام التشغيل

يتم التحكم في الأجهزة المحمولة من خلال أنظمة التشغيل الخاصة بها (OS) بطريقة مشابهة لأجهزة الحاسوب المحمولة وأجهزة الحاسوب. وبالمثل، مثلما تقدم أجهزة الحاسوب المحمولة وأجهزة الحاسوب أنظمة تشغيل مختلفة (Microsoft Windows وLinux وApple OS X)، فإن غالبية سوق الأجهزة المحمولة ينقسم بين تلك الأجهزة التي تعمل بنظام التشغيل Google Android OS (انظر الشكل 7.3) وتلك التي تعمل بنظام التشغيل Apple iOS (انظر الشكل 7.2).

يدعم كلا نظامي تشغيل الأجهزة المحمولة وظائف مماثلة ولكن الطريقة التي يستخدم بها المطور كل وظيفة في أثناء تشغيل التطبيق الخاص به غالبًا ما تكون مختلفة تمامًا، ويمكن أن يختلف هذا أيضًا بين إصدارات نظام تشغيل الجهاز المحمول. على سبيل المثال، من الشائع جدًا أن يقوم إصدار نظام التشغيل الجديد (بإيقاف) تطبيق هاتف محمول كان يعمل سابقًا لأنه غيّر بعض التفاصيل الصغيرة أو الإعدادات.

بالإضافة إلى ذلك، عادةً ما تكون لغة البرمجة المستخدمة لإنشاء التطبيق لكل نظام تشغيل مختلفة وتوفر تحدياتها ومكافآتها ومزاياها وعيوبها. تتم كتابة تطبيقات Android تقليديًا باستخدام Java من Oracle (على الرغم من إمكانية استخدام C أو C++ أيضًا)، بينما يتم إنشاء تطبيقات Apple iOS باستخدام Objective-C.

الشكل 7.2 Apple iOS

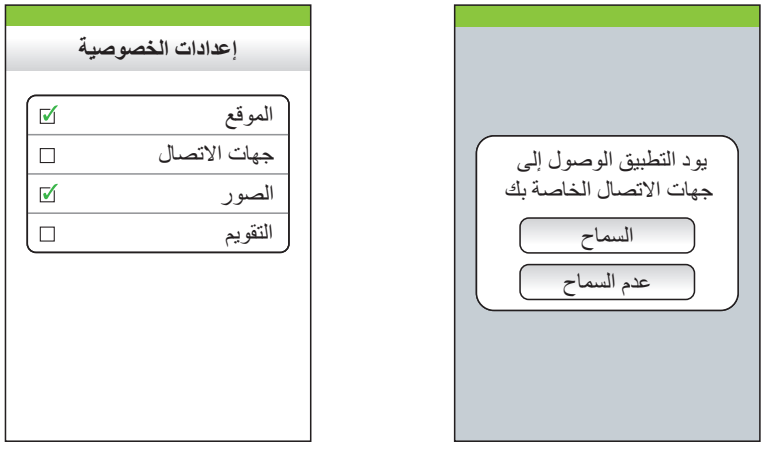
الشكل 7.3 جوجل أندرويد

أذونات الجهاز

غالبًا ما يرغب مطورو تطبيقات الأجهزة المحمولة في الوصول إلى بيانات أو وظائف معينة مضمنة في جهاز محمول لجعل تطبيقاتهم أكثر جاذبية للعملاء. وتأخذ الشركات المصنعة للأجهزة المحمولة الأمان وسلامة أجهزتها على محمل الجد؛ لذلك، غالبًا ما تحتاج التطبيقات إلى الحصول على إذن لاستخدام البيانات والوظائف داخل الجهاز من خلال تدخل المستخدم اليدوي.

ربما تكون قد اختبرت هذا بنفسك. ستظهر رسالة منبثقة تسألك عما إذا كان تطبيق الجهاز المحمول الذي تستخدمه يمكنه الوصول إلى موقعك وقراءة جهات الاتصال أو حالة الهاتف وحتى الوصول إلى شبكتك (انظر الشكل 7.4). قد تحتاج بعض تطبيقات الأجهزة المحمولة، مثل مشغلات البودكاست، إلى إذن لتنزيل البيانات إذا لم تكن متصلة بشبكة لاسلكية، وذلك ببساطة لأن رسوم بيانات شبكة الهاتف المحمول يمكن أن تكون باهظة الثمن.

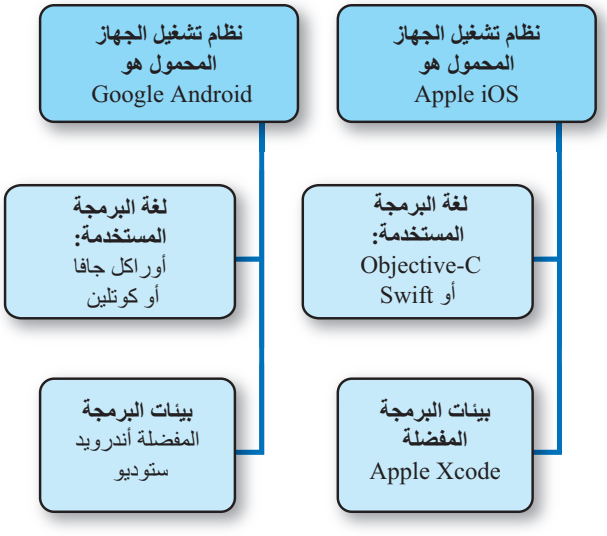
بصفتك مطورًا لتطبيقات الأجهزة المحمولة، يجب ألا تحدد أن يعتمد التطبيق على البيانات أو الوظائف التي قد لا يمنحها المستخدم.



الشكل 7.4 نظام تشغيل يطلب الإذن من المستخدم للتطبيق للوصول إلى معلومات جهات الاتصال الخاصة به

برمجة تطبيقات الأجهزة المحمولة

بصفتك مطورًا لتطبيقات الأجهزة المحمولة، فإن اختياراتك، بما في ذلك لغة البرمجة التي يجب أن تتعلمها وبيئة التطوير المتكاملة التي تنشئ فيها تطبيقك، يتم تحديدها مسبقًا بواسطة نظام تشغيل الجهاز المحمول الذي تنوي إنشاء تطبيقك له، كما هو موضح في الشكل 7.5.



الشكل 7.5 تحديد أدوات التطوير ولغة البرمجة

المصطلحات الرئيسة

مجموعة تطوير البرامج (SDK) - مجموعة من أدوات البرامج المقدمة للمطورين لإنشاء تطبيق لمنصة معينة، على سبيل المثال Java SDK لنظام Android. عادةً ما يحتوي SDK على واجهات برمجة التطبيقات (APIs)، وأدوات مثل المترجمات وأدوات تصحيح الأخطاء، ووثائق مرجعية، ونماذج من الكود.

لغات البرمجة كائنية التوجه – هي نهج حديث في البرمجة (نموذج) لتطوير البرمجيات، يعمل عن طريق نمذجة المشكلات الواقعية وتبسيط العمليات المعقدة إلى تفاعلات أساسية بين الكائنات المختلفة، مثل العلاقة بين العميل وحسابه البنكي.

المهارات

- مهارات التحليل واتخاذ القرار
- اختيار أدوات وأنظمة تكنولوجيا المعلومات المناسبة المستخدمة لتطوير حلول تطبيقات الأجهزة المحمولة

لغات البرمجة

Java هي لغة برمجة قوية وشائعة، عمرها أكثر من 20 عامًا وتستخدم في مليارات الأجهزة حول العالم. تستخدم **مجموعة أدوات تطوير البرامج (SDK)** لنظام Android لغة Java كأساس لبناء تطبيقاته المحمولة. Objective-C، وهي مجموعة شاملة من لغة قديمة تسمى C، موجودة منذ عام 1983 وهي اللغة الرئيسة التي تستخدمها Apple لسطح المكتب OS X وأنظمة تشغيل iOS المحمولة.

على الرغم من كونهما لغات برمجة مختلفة، إلا أن Objective-C وJava هما **لغات برمجة كائنية التوجه** وتتشابهان من حيث هيكلهما وصياغتهما. ونتيجة لذلك، قد تجد أن الإتقان في أحدهما سيكون مفيدًا لتعلم الآخر.

يعد التطوير لنظام Android مجانيًا بشكل أساسي، على الرغم من أن رسوم التسجيل لمرة واحدة مطلوبة لنشر التطبيقات المجانية أو التجارية التي سيتم توزيعها من خلال Google Play. يبدو أن تطوير Apple iOS أكثر تكلفة لأنه يتطلب رسوم تسجيل سنوية.

بحث

قم بزيارة صفحة دعم العضوية في أبل واكتشف خيارات العضوية المختلفة التي تتيح تطوير تطبيقات iOS لأجهزتها المحمولة عبر زيارة: http://developer.apple.com

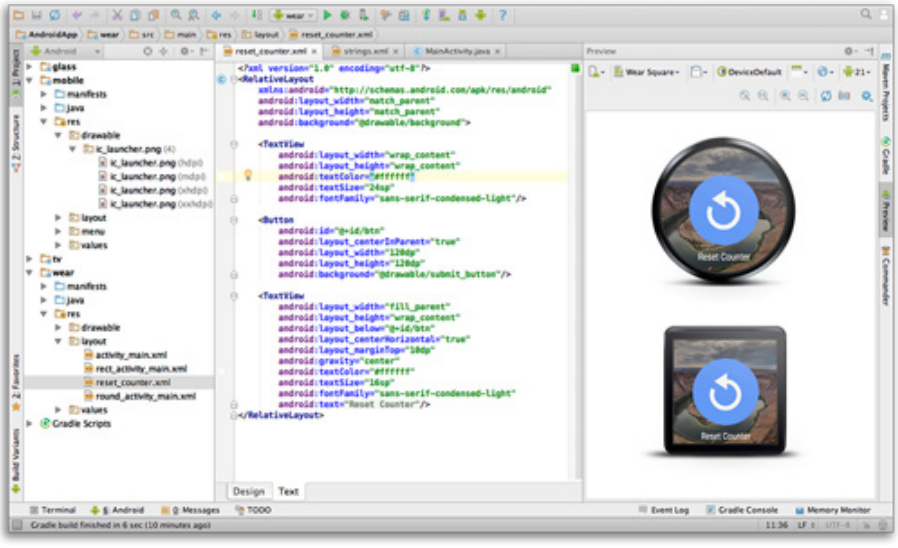
تعرف كيفية تسجيل تطبيقات Android ونشرها من خلال Google Play من خلال زيارة: http://developer.android.com

يركز هذا الكتاب بشكل أساسي على Android وJava، ولكنك ستجد أن إجراء بعض الأبحاث لإجراء مقارنات مفيد جدًا.

بيئات البرمجة

يتمثل دور بيئة البرمجة الحديثة في بيئة تطوير متكاملة (IDE). توافر بيئة التطوير المتكاملة (IDE) للمطورين مجموعة شاملة من الأدوات، بما في ذلك:

- **أدوات إدارة المشاريع** - لإنشاء تطبيقاتك وتنظيمها وإدارتها
- **أدوات التصميم** - لإنشاء واجهات المستخدم، عادةً باستخدام تقنيات (السحب والإسقاط)
- **محرر نصوص مميز بالكامل** - لإدخال الشفرة المصدرية لتطبيقك وتعديلها وحفظها
- **إكمال الكود** - للإكمال التلقائي المفيد لتسريع التطوير
- **مترجم** - لترجمة الكود المصدري لتطبيقك بحيث يعمل على الجهاز المحمول المستهدف
- **تظليل بناء الجملة** - تلوين عناصر مختلفة من كود البرنامج لتحسين قابلية القراءة
- **أدوات التوثيق** - لإضافة تعليقات تشرح كيفية عمل تطبيقك
- **أدوات التصحيح** - للمساعدة في تحديد أخطاء البرنامج وإزالتها
- **أدوات المحاكاة** - لعرض تطبيقك على جهاز محمول افتراضي
- **أدوات الاختبار** - لإظهار أداء تطبيقك واستخدام الموارد وما إلى ذلك.
- **أدوات النشر** - لنقل التطبيق إلى متجره الإلكتروني أو الجهاز المحمول الفعلي.



الشكل 7.6 Android Studio هو بيئة التطوير المتكاملة الرسمية لتطبيقات أندرويد

وقفة للتفكير هل يمكنك شرح نتاج التعلم؟ ما العناصر التي وجدتها أسهل؟

تلميح أغلق الكتاب وقم بعمل قائمة بأنواع وفئات تطبيقات الأجهزة المحمولة المختلفة المتاحة.

توسيع الأفق ما العوامل التي تؤثر على تصميم وميزات تطبيق الجهاز المحمول؟

تمرين تقييمي 7.1 A.P1, A.P2, A.M1, A.D1

على خطاك، سينضم مطور مبتدئ آخر إلى فريق التطوير الخاص بك الأسبوع المقبل. على الرغم من أن لديهم بعض الخبرة في البرمجة، إلا أنهم يتمتعون بخبرة قليلة جدًّا في تطوير تطبيقات الأجهزة المحمولة ولكن من المتوقع أن يتعلموا بسرعة كبيرة.

لقد طلب منك المدير المباشر إعداد وتقديم عرض تقديمي يشرح كيف يتأثر تصميم تطبيق الجهاز المحمول بالغرض المقصود منه واحتياجات وتفضيلات وخصائص المستخدمين المستهدفين. بالإضافة إلى ذلك، طُلب منك تقديم نظرة عامة على تقنيات الأجهزة المحمولة الحالية، مع شرح كيفية تأثيرها على تصميم وتنفيذ مثل هذه التطبيقات.

يجب أن تختتم عرضك التقديمي بمثال عملي يحلل تصميم وتنفيذ بعض نماذج تطبيقات الأجهزة المحمولة، بالنظر إلى هذه العوامل المؤثرة، بحيث يقيّم المثال كيفية تأثير التطبيقات في فعالية التصميم والتنفيذ.

التخطيط

- ما المهمة؟ ما المطلوب مني فعله؟
- ما مدى الثقة التي أشعر بها في قدرتي على إكمال هذه المهمة؟
- هل هناك أي جوانب قد أواجه صعوبة فيها؟

التنفيذ

- أعرف ما أفعله وما أريد تحقيقه.
- يمكنني تحديد متى أخطأت وتعديل تفكيري أو نهجي لإعادة نفسي إلى المسار الصحيح.

المراجعة

- يمكنني شرح المهمة وكيف تعاملت مع المهمة.
- أستطيع أن أشرح كيف سأتعامل مع العناصر الصعبة بشكل مختلف في المرة القادمة (أي ما سأفعله بشكل مختلف).

ب تصميم تطبيق جهاز محمول يستخدم وظائف الجهاز

من أفضل الممارسات تصميم تطبيق الهاتف المحمول الخاص بك بدقة قبل بدء التطوير العملي. في هذا الجزء، ندرس اعتبارات التصميم والإجراءات التي يجب عليك اتخاذها عند تصميم تطبيق الجهاز المحمول الخاص بك.

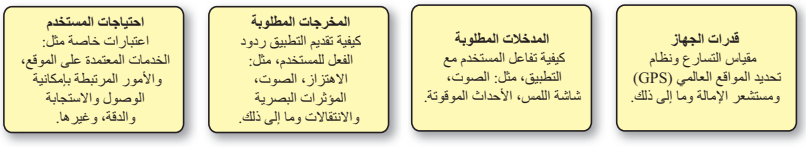
المهارات

- مهارات التحليل واتخاذ القرار
- اختيار أدوات وأنظمة تكنولوجيا المعلومات المناسبة المستخدمة لتطوير حلول تطبيقات الأجهزة المحمولة
- مهارات الإدارة الذاتية والتخطيط

تحليل متطلبات التطبيق

خطوتك الأولى عند تصميم تطبيق هاتف محمول هي مراعاة متطلبات الحوسبة الأساسية. يتعلق بعضها بقدرات الجهاز، بينما يركز البعض الآخر على ما يحتاج التطبيق إلى القيام به (وظائفه الأساسية) والاحتياجات المحددة للمستخدم. يمكننا تقسيم ذلك باستخدام مخطط رباعي، وهو أداة مرئية بسيطة تركز على أربعة مجالات أساسية (انظر الشكل 7.7).

يجب أن يكون من الممكن إكمال رسم تخطيطي رباعي لأي تطبيق هاتف محمول محتمل.

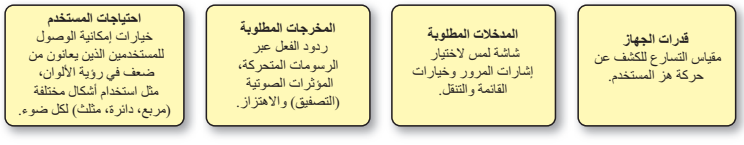


الشكل 7.7 رسم تخطيطي رباعي لمتطلبات التطبيق

مثال عملي: إشارة المرور

هناك حاجة إلى تطبيق هاتف محمول بسيط لتعليم تسلسل الألوان الصحيح لإشارة المرور (جانب إلزامي من اختبارات القيادة). يجب أن يكون المستخدمون قادرين على رؤية الرسوم المتحركة الموقوتة لتسلسل إشارات المرور وأن يكونوا قادرين على لعب لعبة بسيطة حيث يتوقعون الضوء الآتي في التسلسل عن طريق لمس الضوء الصحيح. يجب أن تُستقبل الإجابات الصحيحة بتصفيق تهنئة؛ بينما يجب أن تجعل الإجابات غير الصحيحة الجهاز يهتز. يجب تضمين خيارات إمكانية الوصول للمستخدمين الذين يعانون من نقص في رؤية الألوان. تتم إعادة ضبط إشارات المرور إلى القائمة الرئيسة من خلال إيماءة اهتزاز بسيطة.

- **الخطوة 1:** تحليل وتحديد العناصر لكل فئة من فئات المتطلبات.
- **الخطوة 2:** أكمل مخططًا رباعيًا لتنظيم تفكيرك.



الخطوة 3: تحقق من تحديد جميع العناصر وتصنيفها بشكل صحيح.

تصميم تطبيق جهاز محمول

يعد إنشاء الوثائق المناسبة جزءًا مهمًا وضروريًا من عملية تصميم تطبيقات الأجهزة المحمولة. إذا كنت تتطور بشكل فردي، فهذا يساعد على تنظيم أفكارك وتخطيطك. إذا كنت تعمل ضمن جزء من فريق، فهذا يساعد على توصيل الأفكار ومشاركة حل المشكلات وتحديد المشكلات المحتملة عند ظهورها.

يجب أن تحتوي وثائق التصميم على الحد الأدنى من:

- متطلبات المستخدم الفعلية
- حل مقترح.

متطلبات المستخدم

تحدد متطلبات المستخدم المشكلة التي تحاول حلها. في الواقع، لا ينبغي إجراء أي محاولة لحل المشكلات قبل أن تفهم تمامًا ما يريده المستخدم بالفعل. في بعض الأحيان، من الضروري تضييق نطاق متطلبات المستخدم بناءً على الموارد المتاحة أو تأكيد فهمك لها بشكل أكثر شمولاً، عادةً عن طريق طرح أسئلة إضافية للعميل أو من خلال أبحاث السوق للمستخدمين المحتملين.

الحل المقترح

يمثل الحل المقترح مخططًا لكيفية إنشاء تطبيق الجهاز المحمول. يجب أن تتضمن وثائق التصميم جميع تفاصيل المخطط. من أجل إنشاء مخطط برنامج شامل، هناك العديد من العناصر التي يجب تضمينها. هذه العناصر موضحة في الأقسام أدناه.

وصف مهام البرنامج

وصف مهام البرنامج هو قائمة بالوظائف الأساسية لتطبيق الجهاز المحمول، والتي تم إنشاؤها بناءً على متطلبات المستخدم الفعلية.

لا يلزم إدراج المهام المنفذة بترتيب زمني؛ غالبًا ما يكون من المستحيل تحقيق ذلك حيث يمكن الوصول إلى الوظائف على تطبيق الجهاز المحمول بطرق مختلفة. ومع ذلك، يجب أن تكون شاملة وتفي بمتطلبات المستخدم على أكمل وجه ممكن وأن تكون قابلة للاستخدام كقائمة مرجعية (المهام) بمجرد بدء التطوير الرسمي.

المنصة المستهدفة

يجب أن تحدد المنصة ما يأتي:

- جهاز (أجهزة) الهاتف المحمول المطلوبة
- نظام (أنظمة) التشغيل بما في ذلك الإصدار المستهدف
- نوع التطبيق (أصلي أو مستند إلى الويب أو هجين).

تتضمن الأجهزة المحمولة العديد من الهواتف الذكية أو الأجهزة اللوحية أو التكنولوجيا القابلة للارتداء ولكن قد تحتاج أيضًا إلى تحديد إصدارات معينة. على سبيل المثال، إذا كان تطبيقك يحتاج إلى كاميرا أمامية وخلفية للعمل بشكل صحيح، فقد يقتصر ذلك على طرازات معينة. قد تكون بعض التطبيقات مصممة لأحجام شاشات معينة؛ إذا كانت هذه هي الحالة، فتأكد من أن وثائق التصميم توضح ذلك.

على الرغم من أن تطوير تطبيق الجهاز المحمول الخاص بك سيستهدف بالتأكيد نظام تشغيل معين (مثل Android بدلاً من Apple iOS)، يجب عليك التأكد من أن أي ميزة برمجية رئيسة استخدمتها ليست خاصة بإصدار نظام التشغيل (مثل Android 6.1) ما لم يكن ذلك أمرًا لا مفر منه. سيؤدي القيام بذلك إلى تقييد المستخدمين المحتملين أو إجبارهم على ترقية نظام التشغيل الخاص بهم، وستصبح بعض الأجهزة قديمة جدًا.

المهارات

- مهارات التحليل واتخاذ القرار
- التواصل الكتابي الرسمي
- اختيار أدوات وأنظمة تكنولوجيا المعلومات المناسبة المستخدمة لتطوير حلول تطبيقات الأجهزة المحمولة
- مهارات الإدارة الذاتية والتخطيط
- القدرة على العمل بطريقة قانونية وأخلاقية

المصطلح الرئيس

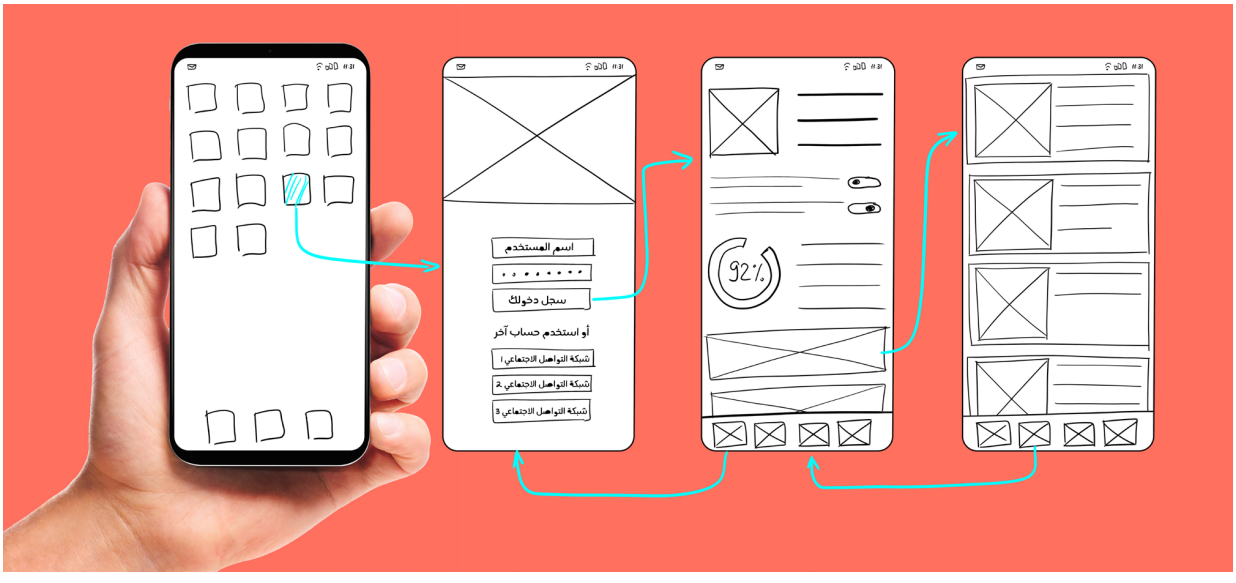
التأطير السلكي - خطوة مهمة في عملية تصميم الشاشة، حيث تساعد المطور على تخطيط الشكل النهائي وتصفح المستخدم باستخدام النماذج الورقية أو الإلكترونية للأجهزة ومكوناتها المرئية.

تخطيطات الشاشة والتنقل

التصميم المرئي هو حجر الزاوية في تطوير تطبيقات الأجهزة المحمولة الجيدة. يتضمن التصميم المرئي مبادئ التخطيط الجيد للشاشة والتنقل البديهي للمستخدم.

يستخدم العديد من المصممين نماذج رسومية للأجهزة، جنبًا إلى جنب مع شاشاتهم وعناصر واجهة المستخدم الخاصة بهم لإنشاء نموذج أولي للتطبيق وتلقي التعليقات قبل كتابة أي سطور من كود البرنامج فعليًا.

تتوفر مجموعات التصميم عبر الإنترنت التي تسمح للمطورين بوضع نماذج أولية لتخطيطات الشاشة والتنقل باستخدام وظيفة (السحب والإسقاط) البسيطة. المصطلح الصناعي لهذه العملية هو **التأطير السلكي** (انظر الشكل 7.8).



الشكل 7.8 أمثلة على الإطارات السلكية لنظام Android

بحث

قم بزيارة نموذج لأداة تأطير سلكي للتعرف على أدوات الإنشاء السريع للنماذج الأولية المرئية لتطبيق الجهاز المحمول الخاص بك. Balsamiq هي إحدى هذه الأدوات وهناك عرض توضيحي مجاني لها متاح عبر الإنترنت. لمشاهدة هذا العرض التوضيحي لعملية التأطير السلكي، قم بزيارة الموقع: www.balsamiq.com

الخوارزميات

الخوارزمية هي ببساطة مجموعة من التعليمات التي يمكن اتباعها لحل مشكلة أو إجراء عملية حسابية.

يمكن إنشاء التطبيقات من العديد من الخوارزميات المختلفة، التي يتم تنفيذها في كود البرنامج باستخدام مزيج من تركيبات البرمجة والوظائف والإجراءات. يمكن تمثيل الخوارزميات المستخدمة في تطبيقات الأجهزة المحمولة باستخدام أدوات تصميم مختلفة مثل:

- **الكود الزائف** - خطوات غير رسمية تشبه اللغة الطبيعية لبرنامج يمكن تحويله إلى لغة البرمجة المستهدفة. لذلك يمكن كتابة الكود الزائف بأي لغة، على الرغم من أن العديد من المطورين يفضلون اللغة الإنجليزية.

- **مخططات النشاط** - تُعرف أيضًا باسم مخططات نشاط لغة النمذجة الموحدة (UML)، وهي توضح تدفق نشاط المستخدم من خلال مجموعة منسقة من الإجراءات، على سبيل المثال استخدام الترميز القياسي لتسجيل الدخول أو شراء عنصر أو حجز موعد
- **المخططات الانسيابية** - تمثيل رسومي للبرنامج، يُظهر أفعاله ومنطقه من خلال مجموعة من الرموز الموحدة.

أدوات التصميم هذه ليست خاصة بتطوير تطبيقات الأجهزة المحمولة. يمكن نقل المهارات التي تقوم ببنائها باستخدامها بسهولة إلى بيئات برمجة أخرى.

موضوعات ذات صلة

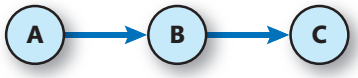
لمزيد من المعلومات حول بيئات البرمجة المختلفة، راجع **الوحدة 6: تطوير المواقع الإلكترونية، والوحدة 8: تطوير ألعاب الحاسوب**.

هياكل التحكم

عادةً ما يتم إنشاء الخوارزميات التي تتحكم في تطبيقات الأجهزة المحمولة باستخدام مزيج من ثلاث كتل أساسية للبرمجة أو هياكل التحكم.

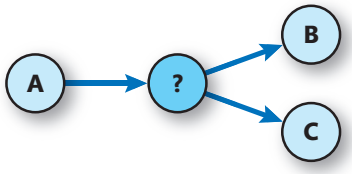
هياكل التحكم هذه هي التسلسل والاختيار والتكرار.

1 التسلسل - إجراء واحد تلو الآخر، لا شيء مفقود، لا يتم تكرار أي إجراء (انظر الشكل 7.9).



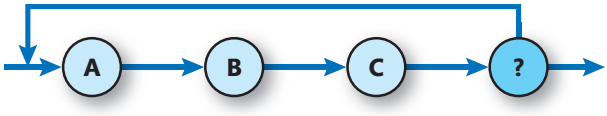
الشكل 7.9 تسلسل

2 الاختيار - الإجراءات التي يتم اختيارها بناءً على حالة تم توفيرها والتي يتم تقييمها على أنها صحيحة أو خاطئة (انظر الشكل 7.10).

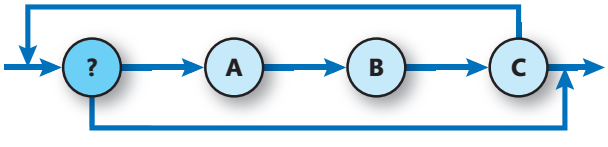


الشكل 7.10 اختيار

3 التكرار - يتم تكرار الإجراءات عدة مرات، عادةً، حتى يصبح الشرط غير قائم (انظر الشكلين 7.11 و7.12).



الشكل 7.11 التكرار المشروط (أو الحلقة)



الشكل 7.12 التكرار المشروط مسبقًا (أو الحلقة)

التحقق من صحة البيانات

عادةً ما تستخدم تطبيقات الأجهزة المحمولة عناصر واجهة المستخدم مثل مفاتيح التشغيل/الإيقاف وأدوات الانتقاء ومربعات القوائم للإدخال، مما يزيل (قدر الإمكان) الحاجة إلى إدخال لوحة المفاتيح التقليدية.

عند حدوث الإدخال المستند إلى النص، فإنه يحتاج عادةً إلى التحقق من صحته. التحقق من الصحة هو ببساطة التحقق من منطقية الشيء قبل معالجته: على سبيل المثال، يجب إدخال (العمر) باستخدام الأرقام فقط.

يُعد (العمر) المُدخل البالغ 21 عامًا مقبولًا، ولكن قيمة F4 أو كتابة (twenty one) ليست كذلك. يعد إنشاء قواعد التحقق، التي تتحقق ما إذا كانت المدخلات المختلفة معقولة في البرنامج أمرًا مهمًا للغاية، حيث يمكن أن يؤدي عدم القيام بذلك إلى نتائج غير دقيقة أو، ما هو أكثر خطورة، **خطأ وقت التشغيل** أو تعطل فادح في التطبيق.

المصطلح الرئيس

خطأ وقت التشغيل - مشكلة تحدث في أثناء استخدام التطبيق، ما يؤدي عادةً إلى قفله (رفض قبول إدخال المستخدم) أو تعطله (إنهاء المستخدم وإعادته إلى قائمة الجهاز أو الشاشة الرئيسة).

تكامل إمكانيات الجهاز

في حالة الحاجة إلى إمكانات جهاز معينة (الوظائف أو جوانب الواجهة أو ميزات نظام التشغيل أو أذونات معينة)، فيجب توثيقها كجزء من التصميم. من المهم معرفة كيف ومتى وأين سيتم استخدامها في تطبيقك.

الحلول البديلة

لمعظم الحلول طرق بديلة لها أيضًا بعض المزايا. يجب أن يكون الجزء المهم من وثيقة التصميم هو تغطية حلول التصميم المختلفة وخطط تسليم التطبيق. وقد تقترب هذه الحلول من اتجاه معاكس، أو تستهدف أجهزة مختلفة أو قد تتطلب موارد أكثر أو أقل (الوقت والمال والخبرة) لإكمالها.

إن وجود حلول بديلة من شأنه أن يضمن استخدامك لجميع الامكانات، والأهم من ذلك، وجود خطة طوارئ في حالة عدم سير الأحداث وفقًا للخطة في أثناء تطوير الحل الذي اخترته.

الموارد والأصول

يجب أن يوضح تصميمك بالتفصيل أي موارد وأصول موجودة تحتاج إلى دمجها في تطبيقك مثل التعليمات البرمجية المحددة مسبقًا (لك و/أو من مكتبة تابعة لجهة خارجية) وأصول الوسائط. قد تتضمن أصول الوسائط الخاصة بك:

- **الصور أو الرسوم** (مثل ملفات الصور.jpg أو.png)
- **الفيديو** (على سبيل المثال مقاطع فيديو.avi أو.mpg أو.mov)
- **الصوت** (مثل ملفات الصوت.wav أو.mp3 أو.au أو.aiff).

احرص على ضمان معالجة أصول الوسائط بشكل مناسب قبل إدراجها. على سبيل المثال، يجب قصها إلى الحجم المناسب و**تحسينها** لتحقيق الكفاءة. يعمل إدراجها ضمن جزء من وثائق التصميم أيضًا كقائمة مرجعية لإعداد المحتوى.

المصطلح الرئيس

محسّنة - يتم إنشاء الأصول المحسّنة باستخدام تنسيقات ملفات أكثر كفاءة لأنها تتطلب مساحة تخزين أقل. يمكن أن يؤدي ذلك إلى تحسينات في الأداء ومساحة رقمية أصغر على موارد الجهاز. من الأمثلة على ذلك استخدام صور .jpg بدلاً من ملفات .bmp حيث تستخدم هذه الأخيرة ضغط البيانات لتقليل حجم الملف.

جدول الاختبار والمراجعة

تعد جدولة الاختبارات والمراجعات القوية جزءًا مهمًا من عملية التصميم. كما سنرى، يعد الإعداد لخطط اختبار شاملة واختيار بيانات الاختبار المناسبة أمرًا ضروريًا لضمان أداء التطبيق الذي قمت بإنشائه بشكل موثوق وكما هو متوقع على الجهاز (الأجهزة) المستهدف.

يتم تحقيق المراجعة على أفضل وجه من خلال تحليل ملاحظات المستخدم المحددة. الهدف من مراجعة تصميم تطبيق الجهاز المحمول الخاص بك هو المساعدة في تحسينه قبل تطويره ثم إصداره رسميًا. من خلال مراجعة التصميم، يجب أن تكون قادرًا على إزالة أي عيوب أو مشاكل قبل التطوير.

القيود

القيود هي العوامل المقيدة التي تتم مواجهتها على المستوى الشخصي أو الجماعي أو التي تفرضها المنصة المستهدفة.

على سبيل المثال، قد تكون مقيدًا بالوقت أو تكاليف التطوير أو التكنولوجيا المتاحة أو ببساطة بخبرتك الفنية الخاصة أو خبرة فريقك.

قد تكتشف أيضًا أن تصميمك مقيد بالمنصة المحددة. قد تجبرك الأذونات والإمكانيات والقيود الخاصة بنظام التشغيل أو الجهاز على اتخاذ قرارات تصميم معينة.

الاعتبارات القانونية والأخلاقية

يجب أن تؤخذ الاعتبارات القانونية والأخلاقية في الاعتبار كجزء من وثائق التصميم الخاصة بك، وخاصة تلك ذات الصلة في بلدك. في حالة تطوير تطبيقات للمستخدمين في بلدان أخرى، ستحتاج إلى مراعاة المتطلبات القانونية والأخلاقية المحددة لتلك الأسواق. من الضروري أحيانًا توفير إصدارات بديلة من التطبيق في مناطق مختلفة. الخصوصية مهمة للمستخدمين، لذلك يجب أن تفكر في كيفية قيام تطبيقك بجمع البيانات الشخصية والتعامل معها والتحكم فيها وتأمينها، وما إذا كان من الضروري للغاية جمع أنواع معينة من البيانات في المقام الأول.

من الناحية الأخلاقية، هناك مخاوف بشأن البيانات التي يمكن مشاركتها من قبل الشركات التي تطور التطبيقات. يمكن استخدام البيانات الشخصية التي يتم جمعها للتأثير في قرارات التأمين أو الائتمان أو التعليم أو التوظيف. كما ذكرنا سابقًا، تعتمد الأجهزة المحمولة الحديثة عمومًا على الأذونات لمنع استخدام البيانات الشخصية للمستخدمين من قبل الشركات التي تمتلك التطبيقات دون موافقتهم، ولكن في النهاية، فإن الأخلاقيات التوجيهية للمطور هي الحماية النهائية.

انتبه إلى المشكلات المحتملة مثل انتهاك حقوق الطبع والنشر، خاصة في ما يتعلق باستخدام أصول الوسائط مثل الصور والفيديو والملفات الصوتية. على الرغم من أنه يمكن دمج أصول الوسائط بسهولة في تطبيق الجهاز المحمول الخاص بك، إلا أنها تحتفظ بحقوق الطبع والنشر لمنشئها الأصلي وتظل ملكيتها الفكرية، وليس ملكك.

بحث

تعرف قواعد الممارسة الموصى بها لمطوري تطبيقات الأجهزة المحمولة في بلدك. يتضمن ذلك أفضل الممارسات حول ضمان خصوصية المستخدمين.

بحث

قم بزيارة موقع إلكتروني محلي في بلدك وتعرف التشريعات المتعلقة بحقوق النشر والتصاميم وبراءات الاختراع.

وقفة للتفكير هل يمكنك شرح نتاج التعلم؟ ما العناصر التي وجدتها أسهل؟

تلميح كيف تحدد متطلبات التصميم للتطبيق؟

توسيع الأفق ما العناصر التي يجب تضمينها في وثائق الحل المقترح؟

ج تطوير تطبيق جهاز محمول يستخدم وظائف الجهاز

المهارات

- اختيار أدوات وأنظمة تكنولوجيا المعلومات المناسبة المستخدمة لتطوير حلول تطبيقات الأجهزة المحمولة
- مهارات الإدارة الذاتية والتخطيط

بمجرد مراجعة تصميم تطبيق الجهاز المحمول وفقًا لمتطلبات المستخدم وقبوله من قبل العميل، يمكنك بدء عملية التطوير المادي. هناك عدد من المراحل المختلفة للعمل من خلالها والتي ستتحداك لتطوير مجموعة مجزية من المهارات العملية والتقنية في تطوير تطبيقات الأجهزة المحمولة.

إعداد المحتوى لتطبيقات الأجهزة المحمولة

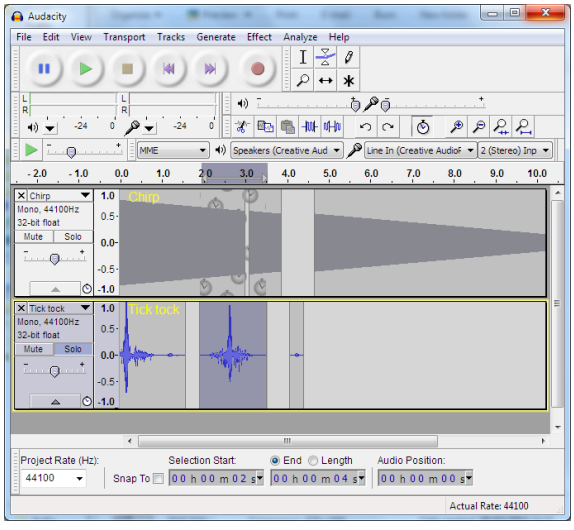
يتم تجميع تطبيقات الأجهزة المحمولة باستخدام أشكال متنوعة من المحتوى الرقمي، بما في ذلك عادةً كود البرنامج والتنسيقات المرئية وملفات الصوت والصور (الرموز والصور والرسوم المتحركة) والفيديو.

قبل إنشاء تطبيق الجهاز المحمول الخاص بك، من الجيد إعداد هذا المحتوى، أي مواردك، بحيث تكون جاهزة لوضعها مباشرة في تطبيقك.

خطوة بخطوة: إعداد المحتوى لتطبيقات الأجهزة المحمولة

5 خطوات

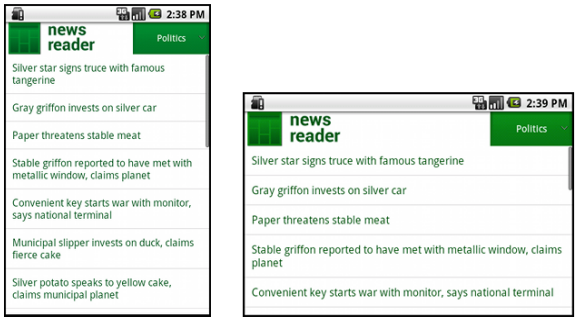
1 حدد التطبيقات والتقنيات المناسبة لإعداد الموارد الخاصة بك (انظر الشكل 7.13).



الشكل 7.13 Audacity® هو مسجل صوت مفتوح المصدر ومجموعة تحرير معروفة.

من المحتمل أن تقتصر خيارات تطوير التعليمات البرمجية الخاصة بك على Android Studio أو Apple Xcode. ومع ذلك، هناك العديد من الخيارات المختلفة لتحرير الصور والملفات الصوتية، بما في ذلك الأدوات المساعدة عبر الإنترنت والبرامج المجانية والبرامج التجارية. اقترحت معظم عمليات التحرير عمليات سير عمل تشجع أفضل الممارسات من أجل تحقيق أفضل النتائج للأجهزة المحددة التي تستهدفها مع تطبيقك.

2 فكّر في كيفية تأثير سمات الجهاز المختلفة على المحتوى الخاص بك عند استخدامه (انظر الشكل 7.14).

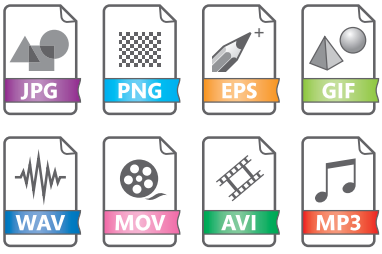


الشكل 7.14 نفس التطبيق على أجهزة ذات أحجام مختلفة وفي اتجاهات مختلفة

تشمل الأشياء التي يجب مراعاتها ما يلي:

- الاتجاه (أفقي أو عمودي أو كليهما)
- حجم الشاشة الفعلي (مثل سبع بوصات)
- دقة الشاشة (بالبكسل) (على سبيل المثال 1024 × 600)
- موارد التطبيق المتاحة (مثل ذاكرة الوصول العشوائي (مثل 1 غيغابايت (GB))
- الصوت (على سبيل المثال، ما إذا كان الجهاز يدعم Dolby)

3 اختر تنسيقات ملفات متوافقة لأصول الوسائط (انظر الشكل 7.15).



الشكل 7.15 تنسيقات ملفات مختلفة - قد لا تكون جميعها مدعومة من قبل الجهاز المستهدف

قد يدعم تطوير التطبيقات (وبعض لغات البرمجة) الملفات بتنسيق معين فقط (على سبيل المثال، قد تكون الصور .jpg أو .png فقط). تأكد من أن أصول الوسائط الخاصة بك متوافرة بالتنسيق الصحيح. إذا لم تكن كذلك، فيمكنك عادةً تحويلها باستخدام برامج متخصصة أو أدوات عبر الإنترنت. يجب عليك أيضًا مراعاة تأثيرات تفاعل المستخدم مع الأصول. على سبيل المثال، سيواجه المستخدم الذي يقوم بتكبير صورة نقطية ظهور الصورة (قاتمة) أو منقطة بشكل مفرط، في حين أن الصور المتجهة (إذا كانت مدعومة) لا تعاني من هذه المشكلة.

4 قم بتحسين المحتوى بحسب الاقتضاء (انظر الشكل 7.16).

الشكل 7.16 تحسين صورة عن طريق اقتصاصها

قم بتطبيق التحسين المعقول على:

- كود البرنامج (على سبيل المثال، إزالة الأقسام غير الضرورية من الكود المكتوب مسبقًا ('boilerplate') والتي توفرها بيئات التطوير المتكاملة (IDE) غالبًا كنقطة بداية مفيدة)
- تقليل أحجام ملفات أصول الوسائط باستخدام تقنيات الضغط مثل حفظ الصور كملفات.png أو .jpg أو تحديد التنسيق الأكثر كفاءة لنوع معين من المحتوى أو قص أجزاء من الملفات غير المطلوبة.

يجب أن تتذكر دائمًا أن هناك مفاضلة بين مستويات التحسين والجودة (اختر ما تفضل وستتأثر جودة المحتوى الخاص بك).

5 فكر في الأمان.

قد تحتوي بعض البيانات أو الأصول على معلومات حساسة. فكّر في استخدام أدوات التشفير لحمايتها وحماية المستخدمين المستهدفين لتطبيقك من الهجمات الإلكترونية. عادةً ما يكون التشفير عبارة عن خوارزمية حسابية تعمل على تشويش البيانات الحساسة باستخدام مفتاح محدد من قبل المستخدم. إذا كانت البيانات مشفرة، فإنها تعتبر آمنة من متطفل الطرف الثالث لأنه لا يمكن إلغاء تشفيرها بشكل صحيح دون مفتاح المستخدم.

نصائح

حافظ دائمًا على مفاتيح المستخدم (كلمات المرور) آمنة وقم بتغييرها بانتظام. حتى البيانات المشفرة قد تكون في خطر إذا لم تكن كلمات المرور آمنة.

موضوعات ذات صلة

لمزيد من المعلومات حول الرسوم (الصور النقطية والمتجهية) راجع قسم *تطوير ألعاب الحاسوب (مثال عملي)*، الصفحة 49 في **الوحدة 8: تطوير ألعاب الحاسوب**.

المهارات

- مهارات التحليل واتخاذ القرار
- التواصل الكتابي الرسمي
- مهارات الإدارة الذاتية والتخطيط
- القدرة على العمل بطريقة قانونية وأخلاقية

فكِّر مليًّا

ينطوي تصميم أي تطبيق هاتف محمول على أعمال تخطيط كبيرة، لا سيما في ما يتعلق بالتحقيق وفهم متطلبات المستخدمين المستهدفين وتعقيدات المنصات المستهدفة. يجب ألا يقتصر أي حل مقترح على سرد الإجراءات التي تحتاج إلى تنفيذها فحسب، بل يجب أن يتضمن تقديرًا لمقدار الوقت الذي تتطلبه كل خطوة للتطوير.

إن تسجيل النتائج بدقة وشمولية خلال مراحل التصميم وإعداد المحتوى يعزز حل مشكلتك من خلال توفير أساس متين يمكن أن يبدأ التطوير الفعال عليه.

تطوير تطبيق جهاز محمول

سيأخذك هذا الجزء خلال عملية تطوير تطبيق الجهاز المحمول لتلبية متطلبات المستخدم المحددة، وضمن جزء من هذه الرحلة، سيقدم لك المفاهيم الأساسية التي تشكل لغة برمجة الجهاز المحمول وبيئة التطوير المستخدمة في بنائها. يستخدم هذا المثال لغة برمجة Android Studio وبرمجة جافا.

سيوضح لك المثال الآتي كيفية تصميم تطبيق بسيط لتحويل درجة الحرارة يسمح للمستخدم بالتحويل بين القراءات بالدرجة المئوية والدرجة فهرنهايت. يمكن رؤية التصميم الأساسي في الشكل 7.17.

لإنشاء هذا التطبيق، ستحتاج إلى تنزيل وتثبيت Java SDK ونسخة من Android Studio، وكلاهما متاح مجانًا. يجب تثبيت Java SDK أولًا، متبوعًا بنظام Android Studio.

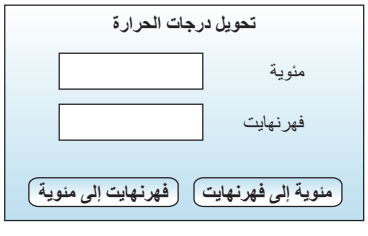
فكِّر مليًّا

مهارات الاتصال الجيدة ضرورية للتحقيق في متطلبات المستخدم المستهدف وتسجيلها. قد يشمل ذلك التواصل الكتابي والشفوي، ويتضح ذلك من خلال أنشطة مثل إجراء المقابلات الشخصية وإرسال رسائل البريد الإلكتروني والرد عليها وإنشاء وثائق تصميم رسمية لتطبيق الجهاز المحمول.

غالبًا ما يعرف المستخدمون المستهدفون ما يريدون من حيث المنتج ولكن لديهم القليل جدًا من الوعي الفني بعملية تطوير تطبيقات الأجهزة المحمولة. سيتعين عليك التفكير وتعلم كيفية ضبط تعبيراتك لتجنب استخدام المصطلحات غير الضرورية والعثور على اللغة والمستوى التقني الأنسب للجمهور المقصود.

بمجرد بدء تشغيل Android Studio وإنشاء مشروع جديد، تظهر لك عادةً نافذة مشروع جديدة (على اليسار). إذا لم يكن الأمر كذلك، فحدد علامة التبويب الرأسية **للمشروع** ثم قم بتوسيع (res) ثم (التخطيط).

يجب أن يؤدي النقر على إدخال content_main.xml إلى عرض تطبيقك الفارغ مع TextView الافتراضي 'Hello World!' في اللوحة المركزية، كما هو موضح في الشكل 7.18.

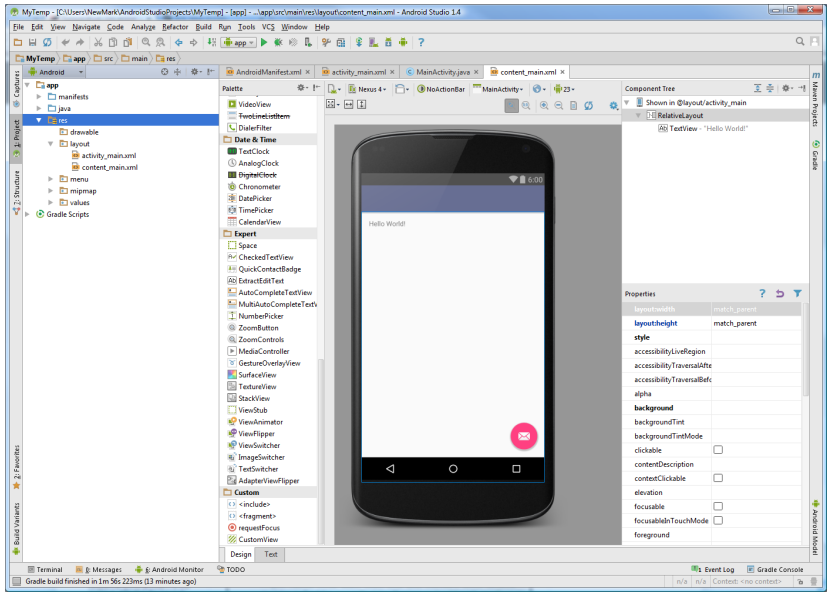


الشكل 7.17 تصميم بسيط لتطبيق تحويل درجة الحرارة

موضوعات ذات صلة

قم بتنزيل Java SDK من www.oracle.com

قم بتنزيل أAndroid Studio من https://developer.android.com



الشكل 7.18 عرض التصميم في Android Studio، يعرض TextView الافتراضي 'Hello World!' TextView

تعمل طريقة عرض تصميم Android Studio على مبدأ (السحب والإسقاط) البسيط، لذلك يجب أن تكون قادرًا على نقل TextView المحدد بحرية في النموذج الرئيس للتطبيق.

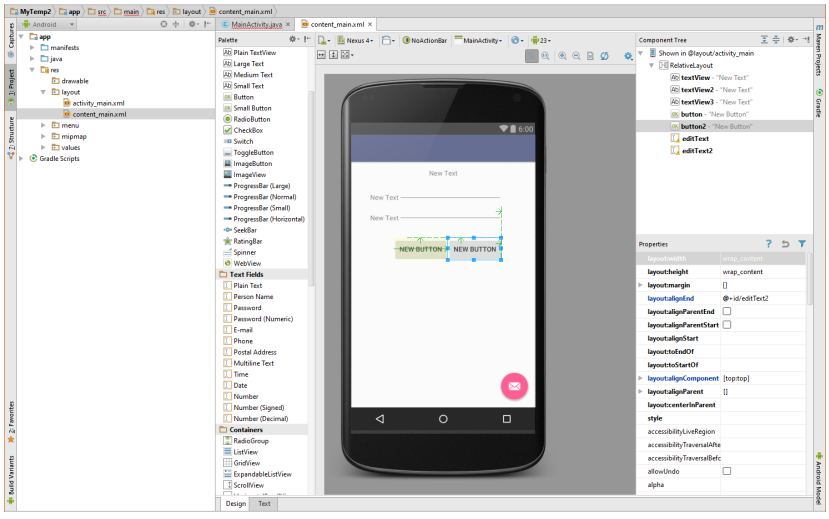
يحتوي Android Studio على عدد من الفئات المختلفة التي يمكن استخدامها لوضع الكائنات على النموذج الرئيس للتطبيق. تظهر هذه في **لوحة الألوان (Palette)** على يسار لوحة التصميم المركزية. يتم تجميع الفئات في الأنواع الآتية:

- **التخطيطات (Layouts)** - التحكم في كيفية تنظيم محتويات النموذج (على سبيل المثال في جدول، في مصفوفة، في صفوف).
- **الأدوات (Widgets)** - أنواع مختلفة من عناصر النماذج المستخدمة لإنشاء واجهة التطبيق، مما يسمح للمستخدم بإدخال البيانات وإجراء التحديدات ومشاهدة المخرجات (مثل وحدات TextViews وButtons وCheckBox).
- **حقول النص (Text Fields)** - أنواع محددة من TextView لوظائف محددة (مثل كلمة المرور وعنوان البريد الإلكتروني ورقم الهاتف).
- **الحاويات (Containers)** - طرق تجميع عناصر النموذج معًا.
- **التاريخ والوقت (Date & Time)** - أنواع مختلفة من عناصر النموذج المتعلقة بتقويم الجهاز والساعة (مثل DataPicker وTimePicker وCalendarView وTextClock وAnalogClock).
- **خبير (Expert)** - أنواع معقدة من عناصر النموذج لمطور التطبيقات الأكثر تقدمًا.
- **مخصص (Custom)** - فئة متخصصة تم إنشاؤها بواسطة المطور أو أداة تابعة لجهة خارجية.

عندما تقوم بسحب وإسقاط فئة من **لوحة الألوان** إلى النموذج، تقوم Java بإنشاء كائن صلب (مثيل ملموس لتلك الفئة) له اسم وخصائص (أشياء تصفه) وأساليب (أشياء يمكنه القيام بها). يقوم Android Studio أيضًا بإنشاء ملف XML (لغة الترميز القابلة للتوسيع) الذي يصف مظهر التطبيق في أثناء إجراء كل تغيير وإضافة؛ ويمكن عرض هذا في علامة تبويب **Text** أسفل الشاشة.

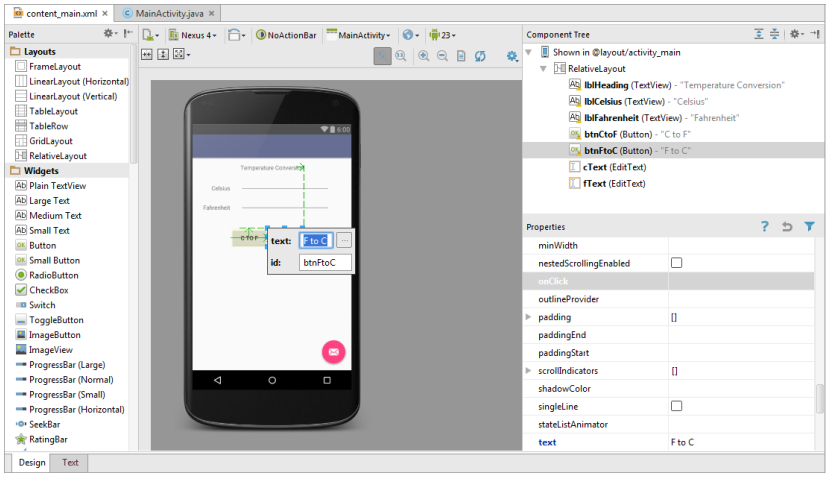
ستقوم الآن بحذف نص TextView "Hello World!" واستبدله بالمحتوى بناءً على تصميم تطبيق تحويل درجة الحرارة. يوضح الشكل 7.19 هذه الإجراءات.

قم بتحرير خصائص النموذج بحيث يشبه التصميم الموضح في الشكل 7.19.



الشكل 7.19 النموذج الرئيس مع كائنات Java الإضافية التي تم سحبها من لوحة Android Studio

يمكن تحرير كل كائن عن طريق تغيير خصائصه (لاحظ نافذة **الخصائص (Properties)** في الزاوية اليمنى السفلية من بيئة التطوير المتكاملة Android Studio). **الخصائص** تتحكم في المظهر المرئي لكل كائن وتتيح لك تعديل حجمه ونوع الخط واللون. من الممكن أيضًا النقر نقرًا مزدوجًا فوق كائن النموذج لتحرير النص والمعرف الخاص به.



الشكل 7.20 عرض تصميم Android Studio، يعرض كائنات النموذج بخصائص ومعرفات جديدة

ربما لاحظت أن كل كائن في النموذج الرئيس لتطبيقك له اسم افتراضي (أو معرف)، على سبيل المثال TextView وTextView2 وEditText وEditText2. على الرغم من فائدة ذلك، إلا أنه لا يولد كود برنامج قابل للقراءة، لذا من الجيد إعادة تسميتها بشكل معقول. قم بذلك عن طريق النقر المزدوج على كل كائن؛ سيظهر مربع حوار صغير يسمح لك بتغيير معرفاتهم، كما هو موضح في الشكل 7.20.

تأكد من تطابق المعرفات الجديدة مع قائمة الكائنات، كما هو موضح في نافذة **شجرة المكونات** في الشكل 7.20.

هياكل البرمجة

توقف الآن وركز على لغة برمجة Java التي تستخدم في كتابة تطبيقات Android.

أوامر الإدخال والإخراج

يتفاعل المستخدمون مع البرنامج من خلال الاستجابة لمطالبات الإدخال. سينتظر البرنامج إدخال المستخدم ويخزن المدخلات في المتغير المحدد. على سبيل المثال، يعرض أمر System.out.print الخاص بلغة Java على السطر 10 (كما هو موضح في الشكل 7.23) طلب تعليمات الشاشة و keyboard.NextInt() فورًا بعد انتظار الإدخال من لوحة المفاتيح المخزنة في الخيار.

في Java، يتم تحقيق الإخراجات إلى الشاشة بنفس الطريقة، ولكن دون انتظار الضغط على المفاتيح الآتية. تستخدم اللغات الأخرى كلمات رئيسة مختلفة. في C++ يكون cin >> (للإدخال) أما cout << (للإخراج)

موضوعات ذات صلة

لمزيد من التفاصيل حول لغة البرمجة، انظر **الوحدة 4: البرمجة**.

الكلمات المحجوزة والمتغيرات المحلية والعالمية والثوابت والتعيين

تحتوي Java، مثل معظم لغات البرمجة، على عدد من الكلمات المحجوزة. الكلمات المحجوزة هي أجزاء أساسية من اللغة التي *لا يمكنك* استخدامها لتسمية الأشياء. توجد قائمة بكلمات Java المحجوزة الشائعة في الجدول 7.4.

الجدول 7.4 قائمة أبجدية بالكلمات المحجوزة في Java (بعضها خاص بإصدارات معينة)

abstract	continue	for	new	switch
assert	default	goto	package	synchronized
boolean	do	if	private	this
break	double	implements	protected	throw
byte	else	import	public	throws
case	enum	instanceof	return	transient
catch	extends	int	short	try
char	final	interface	static	void
class	finally	long	strictfp	volatile
const	float	native	super	while

عندما تقوم بتسمية الأشياء، فإنك تقوم بإنشاء معرفات. هناك نوعان من المعرفات الأساسية - المتغيرات والثوابت. كما تشير الأسماء، تقوم المتغيرات بتخزين القيم التي يمكن لتطبيقك معالجتها. تقوم الثوابت بتخزين القيم التي لا يمكن تغييرها في أثناء تشغيل التطبيق.

مناقشة

غالبًا ما تستخدم تطبيقات الأجهزة المحمولة العديد من المتغيرات والثوابت. تخيل إنشاء تطبيق سهل الاستخدام لحساب أجور وظيفة دوام جزئي. في مجموعة صغيرة، ناقش القيم التي سيتم تخزينها في المتغيرات والقيم التي ستكون ثوابت، مع التأكد من تبرير قراراتك.

من أجل الإعلان عن متغير، يلزم تحديد اسمه و**نوع البيانات** الخاص به. يجب على المطورين تحديد النوع الأنسب للبيانات التي يرغبون في تخزينها. يوضح الجدول 7.5 أنواع بيانات Java البدائية وأمثلة للبيانات المختلفة.

في كل إعلان نقول:

القيمة = اسم متغير نوع البيانات

يتم استخدام علامة = (أو عامل التعيين؛ انظر الجدول 7.5) لتخزين القيمة في المتغير المسمى. المرة الأولى التي تقوم فيها بذلك تسمى (التهيئة)؛ بعد ذلك تسمى ببساطة (المهمة). قد تغير المتغيرات قيمتها عدة مرات في أثناء تشغيل التطبيق الخاص بك.

الجدول 7.5 أنواع بيانات Java البدائية وميزاتها وأمثلة البيانات

نوع البيانات	ما هي	الحد الأدنى - الحد الأقصى للنطاق (عند الحاجة)	القيمة الافتراضية	مثال عن لإعلان
byte	عدد صحيح مكمل مكون من ثمانية بتات	-128 إلى 127	0	byte age = 17;
short	عدد صحيح مكون من 16 بت ومميز بعلامة بنظام المكمل الثنائي	-32,768 إلى 32,767	0	short qty = 10000;
int	عدد صحيح مكون من 32 بت ومميز بعلامة بنظام المكمل الثنائي	-2,147,483,648 إلى 2,147,483,647	0	int largeqty = 100000;
long	عدد صحيح مكون من 64 بت ومميز بعلامة بنظام المكمل الثنائي	-9,223,372,036,854,775,808 حتى 9,223,372,036,854,775,807	0L	long vlargeqty = 100000L;
float	عدد عشري عائم بدقة أحادية مكون من 32 بت وفق معيار IEEE 754		0.0f	float wages = 2814.20f
double	عدد عشري عائم بدقة مزدوجة مكون من 64 بت وفق معيار IEEE 754		0.0d	float salary = 28140.20d
boolean	حالة منطقية ذات 1 بت؛ صحيح أو خطأ	صحيح أو خطأ	خطأ	boolean alive = true;
char	حرف Unicode مكون من 16 بت	'\u0000' to '\uffff'		char initial = 'A';

معظم لغات البرمجة لديها مفهوم المتغيرات المحلية والعالمية. أسهل طريقة للتفكير في ذلك هي تذكر أنه يمكن استخدام المتغيرات العالمية في **أي مكان** في كود التطبيق الخاص بك. **تقتصر** المتغيرات المحلية على استخدامها في مجموعة التعليمات البرمجية التي يتم الإعلان عنها فيها. ومع ذلك، تقوم Java بالأشياء بشكل مختلف قليلاً، كما سترى.

العوامل

العوامل هي رموز خاصة تستخدم لأداء مهام خاصة في معظم التطبيقات. إذا كنت بالفعل على دراية بالرموز مثل استخدام* للضرب، فأنت على دراية بالعامل الحسابي. تحتوي Java على عدد من هذه الأشياء ومن الجيد أن تتعرفها. يتم عرض العوامل الأكثر استخدامًا في Java في الجدول 7.6.

المصطلح الرئيس

نوع البيانات - نوع البيانات التي نريد تخزينها في ذاكرة الوصول العشوائي للجهاز المحمول. يتطلب تخزين كل نوع من البيانات كميات مختلفة من ذاكرة الوصول العشوائي. على سبيل المثال، يختلف نوع البيانات المطلوب لتخزين رقم عن النوع المستخدم لتخزين حرف وسيتطلب أحجامًا مختلفة من ذاكرة الوصول العشوائي. تختلف الأسماء الدقيقة لأنواع البيانات بين لغات البرمجة المختلفة، لذا احرص على اختيار اللغة الصحيحة.

موضوعات ذات صلة

لمزيد من المعلومات حول ذاكرة الوصول العشوائي (RAM)، راجع قسم *الذاكرة والتخزين*، الصفحات 14-16 في **الوحدة 8: تطوير ألعاب الحاسوب**.

المصطلح الرئيس

العامل - رمز خاص (أو رموز متعددة) يخبر البرنامج بإجراء عمليات حسابية أو ارتباطية أو منطقية محددة على بياناته. يجب استخدام العوامل بترتيب أولوية محدد (يصف ذلك ما يتم تنفيذه أولًا). قد تكون معتادًا على هذا المفهوم من استخدام BIDMAS (أو BODMAS) (الأقواس، الأسس/الترتيب القسمة، الضرب، الجمع والطرح) في الرياضيات.

الجدول 7.6 العلميات الحسابية والمنطقية Java الشائعة، مع توضيح مجموعاتها والغرض منها

العمليات الحسابية والمنطقية Java	الاسم	مجموعة المشغل	الغرض
+	جمع	كود حسابي	يضيف قيمتين مع إعطاء مجموعهما
-	طرح	كود حسابي	يطرح قيمة واحدة من أخرى مع إعطاء الفرق بينهما
*	ضرب	كود حسابي	يضرب قيمتين مع إعطاء حاصل الضرب
/	قسمة	كود حسابي	يقسم قيمة واحدة على أخرى مع إعطاء حاصل القسمة
++	إضافة	كود حسابي	يزيد القيمة بمقدار 1
--	نقصان	كود حسابي	يقلل القيمة بمقدار 1
==	يساوي	نسبي	الاختبارات التي تتساوى فيها قيمتان
=!	لا يساوي	نسبي	يختبر ما إذا كانت قيمتان غير متساويتين
<	أكبر من	نسبي	يختبر ما إذا كانت إحدى القيم أكبر من الأخرى
>	أقل من	نسبي	يختبر ما إذا كانت إحدى القيم أقل من الأخرى
>=	أكبر من أو يساوي	نسبي	يختبر ما إذا كانت إحدى القيم أكبر من أو تساوي قيمة أخرى
<=	أقل من أو يساوي...	نسبي	يختبر ما إذا كانت إحدى القيم أقل من أو تساوي قيمة أخرى
=	تعيين بسيط	تعيين	يقوم بتعيين قيمة لمتغير، لا يجب الخلط بينه وبين علامة المساواة المزدوجة التي تختبر المساواة
&&	And	منطقي	يقوم بإجراء عملية AND المنطقية
||	Or	منطقي	يقوم بإجراء عملية Or منطقية
!	Not	منطقي	يقوم بإجراء عملية Not منطقية

بحث

لا يمكن إدراج جميع رموز تشغيل Java هنا. قم بزيارة قسم Java من www.tutorialspoint.com لمعرفة المزيد عن رموز التشغيل الأخرى الموجودة واجمع أمثلة على استخدامها.

تسلسلات التحكم

تذكر هياكل التحكم الثلاثة التي تم تقديمها سابقًا: التسلسل والاختيار والتكرار.

عندما تقوم بإنشاء تطبيق هاتف محمول، يتم تمثيل **التسلسلات** بأي كتلة من الأكواد البرمجية لبرنامج Java

بحث

يمكنك تجربة نماذج الأكواد البرمجية المعروضة بسرعة من خلال زيارة محول Java مجاني عبر الإنترنت. عند إدخال رمز البرنامج الذي تريد تجربته، ما عليك سوى النقر على خياري (التحويل البرمجي) و(التنفيذ). يمكنك حفظ الرمز الخاص بك للاستخدام في المستقبل.

قم بزيارة www.tutorialspoint.com لتجربة ذلك واكتشاف تقنيات ترميز جديدة، على سبيل المثال عبارات if المتداخلة وعوامل التشغيل الشرطية.

موضوعات ذات صلة

لمزيد من المعلومات حول التسلسل والاختيار والتكرار، راجع الجزء الخاص بهياكل *التحكم*، الصفحة 17.

التي تنفذ سطرًا بعد سطر، مع عدم تفويت أي شيء أو تكراره.

يتم إنشاء **الاختيارات** باستخدام إما if... else وإما عبارة switch. يعتمد النوع الذي تستخدمه على ما تحاول تحقيقه.

تحتوي عبارة if... else البسيطة على شرط يتم تقييمه إلى الصواب أو الخطأ. كما هو موضح في الشكل 7.21، يتم تنفيذ الإجراءات في الكتلة الأولى إذا كان الشرط صحيحًا، وإلا يتم استخدام الإجراءات في الكتلة الأخرى (الاختيارية). يظهر الناتج من عمليتي اختبار لكود Java هذا في الشكل 7.22.

```java
import java.util.Scanner;

public class NumberTest {

    public static void main(String []args) {

        int number1;
        int number2;

        Scanner keyboard = new Scanner(System.in);
        System.out.print("Enter 1st number: ");
        number1 = keyboard.nextInt();
        System.out.print("Enter 2nd number: ");
        number2 = keyboard.nextInt();

        if (number1 == number2) {
            System.out.println("Numbers are the same!");
        }
        else {
            System.out.println("Numbers are NOT the same!");
        }
    }
}
```

الشكل 7.21 عبارة جافا if... else

```
Enter 1st number: 4
Enter 2nd number: 5
Numbers are NOT the same!

Enter 1st number: 78
Enter 2nd number: 78
Numbers are the same!
```

الشكل 7.22 الناتج من اختبارين مختلفين لبيان Java if... else

عبارة Java switch هي طريقة مفيدة للتحقق من القيم المتعددة في نفس الوقت، كما هو موضح في الشكل 7.23. من الممكن أيضًا إضافة فحص افتراضي في حالة عدم مطابقة الخيارات المدرجة. يتم عرض المخرجات من أربع عمليات اختبار لجملة switch في الشكل 7.24.

نصائح

العديد من لغات البرمجة الحديثة حساسة لحالة الأحرف وJava ليست استثناء. إذا كنت تواجه مشكلات في تجميع برامجك، فتحقق من أنك استخدمت الأحرف الكبيرة والصغيرة بشكل صحيح في التعليمات البرمجية. يعد فقدان الرموز المهمة مثل النقاط النصفية (التي تُستخدم في Java لفصل العبارات) أيضًا خطأ شائعًا، حتى بالنسبة للمطورين ذوي الخبرة.

```java
import java.util.Scanner;

public class BankAccount {

    public static void main(String []args) {
        int option;

        Scanner keyboard = new Scanner(System.in);
        System.out.println("1 - Bank Balance, 2 - Account Query, 3 - Payments");
        System.out.print("Enter option 1, 2 or 3: ");
        option = keyboard.nextInt();
        switch (option) {
            case 1: System.out.println("You have chosen to see your balance");
                    break;
            case 2: System.out.println("You have chosen to make a query");
                    break;
            case 3: System.out.println("You have chosen to make a payment");
                    break;
            default:System.out.println("You have not chosen a correct option!");

        }
    }
}
```

الشكل 7.23 جملة switch الخاصة بلغة Java

```
You have chosen to see your balance

1 - Bank Balance, 2 - Account Query, 3 - Payments
Enter option 1, 2 or 3: 2
You have chosen to make a query

1 - Bank Balance, 2 - Account Query, 3 - Payments
Enter option 1, 2 or 3: 3
You have chosen to make a payment

1 - Bank Balance, 2 - Account Query, 3 - Payments
```

الشكل 7.24 المخرجات من أربع عمليات اختبار لجملة Java's switch

يتم تمثيل **التكرارات** (تسمى أحيانًا الحلقات) بواحدة من ثلاث بيانات شائعة (for، while and do...while). الأشكال 7.25 إلى 7.27 تعرض حلقات for وwhile وdo…while في Java لمثال معين الشكل 7.28 يوضح أن الإخراج من جميع الحلقات الثلاثة متطابق.

```java
public class ForLoop {

    public static void main(String[]args) {
        int counter = 0;

        for (counter = 1; counter <= 5; counter++) {
            System.out.println("BTEC International");
        }
    }
}
```

الشكل 7.25 حلقة for في Java

```java
public class WhileLoop {

    public static void main(String[]args) {
        int counter = 1;

        while (counter <= 5) {
            System.out.println("BTEC International");
            counter++;
        }
    }
}
```

الشكل 7.26 حلقة while في Java

```java
public class DoWhileLoop {

    public static void main(String[]args) {
        int counter = 1;

        do {
            System.out.println("BTEC International");
            counter++;
        } while (counter <= 5);
    }
}
```

الشكل 7.27 جملة do…while في Java

```
BTEC International
BTEC International
BTEC International
BTEC International
BTEC International
```

الشكل 7.28 الإخراج من جميع الحلقات الثلاث متطابق

المصطلحات الرئيسة

الدالة أو الإجراء - هي مجموعة من التعليمات البرمجية، يفضل أن تكون بين 5 و 50 سطرًا، لها غرض محدد واحد. على الرغم من كتابتها مرة واحدة فقط، يمكن تنفيذها عدة مرات خلال برنامج باستخدام "استدعاء الدالة"، مما يقلل من الحاجة إلى تكرار التعليمات البرمجية. في بعض لغات البرمجة، يتم استخدام مصطلحي "الدالة" و"الإجراء" بالتبادل، ولكن في البعض الآخر، هما مفهومان مختلفان للغاية: عادةً ما تعيد الدالة قيمة محسوبة، بينما يقوم الإجراء بمهمة واحدة قابلة لتعرفها.

كائنية التوجه - جافا، التي نستخدمها لإنشاء تطبيقات أندرويد، مصنفة ضمن لغات البرمجة الكائنية التوجه (OOP) القائمة على الفئات. في لغات البرمجة الموجهة للكائنات، يتم إنشاء الكائنات من فئات يتم تصميمها عادةً على غرار (الأشياء) الواقعية. تعمل كل فئة كخريطة برمجية، حيث تقوم بتغليف (أو احتواء) حالة الشيء (بياناته أو خصائصه) وسلوكياته (وظائفه أو طرقه) في كود البرنامج.

نصائح

يعد تطبيق تحويل درجة الحرارة بسيطًا نسبيًا ويجب أن يستخدم بنية التحكم في التسلسل الأساسي. عندما تبدأ في تصميم تطبيقات أكثر تعقيدًا، ستصبح التركيبات الأخرى (التسلسل والتكرار) مهمة، لذا تذكر عبارات if وswitch وfor and while الموضحة هنا.

على الرغم من أن الحلقات الثلاث تولد نفس المخرجات، إلا أنها عملت جميعًا بشكل مختلف. من الناحية المثالية، يمكنك استخدام الحلقات عندما تعلم أن الكود يحتاج إلى تكرار عدد ثابت من المرات. بينما قد لا يتم تشغيل الحلقات مرة واحدة (فهي مشروطة مسبقًا) وتقوم بذلك... بينما تتكرر الحلقات دائمًا مرة واحدة على الأقل (يتم تكييفها لاحقًا).

تتكرر الحلقات بناءً على صحة حالة التحكم الخاصة بها. في المثال الموضح في الأشكال 7.25 إلى 7.28، يكون شرط التحكم هو متغير العداد الذي له قيمة أقل من (أو تساوي) 5. عندما يصل العداد إلى 6، لا تكون الحالة صحيحة وتنتهي الحلقة.

الدوال والإجراءات

عنصر رئيس آخر لتطبيقات الجهاز المحمول هو مفهوم الدالة أو الإجراء. تعد **الدوال** و**الإجراءات** من السمات الشائعة في البرمجة الإجرائية أما في البرمجة **كائنية التوجه**، فيتم تمثيلها بشكل أساسي بالطرق التي تكون جزءًا من الفئة.

العناصر والفئات

تعد الفئات جانبًا أساسيًا لكل حل من حلول تطبيقات Android للأجهزة المحمولة تم إنشاؤه باستخدام Java. يتم استخدام الفئات لتمثيل المكونات المادية التي تستخدمها لإنشاء تطبيق. على سبيل المثال، يمثل كل عنصر في **لوحة** Android Studio (مكونات الشاشة مثل النماذج والتخطيطات والأدوات وحقول النص) فئة مختلفة يمكنك استخدامها.

بمجرد سحب الفئة إلى النموذج، يتم إنشاء مثيل ملموس لتلك الفئة، والذي نسميه كائنًا. حاول التفكير في الفئة كنمط تصميم، أو شيء مثل قالب الجيلي أو الاستنسل أو قطاعة البسكويت؛ يمكن استخدام قالب جيلي واحد (الفئة) لصنع العديد من قطع الجيلي (الأشياء).

يحتوي كل كائن على أساليب وخصائص مرتبطة ويمكنه التفاعل مع الكائنات الأخرى لتنفيذ مهمة. تستخدم التطبيقات التجارية مجموعة واسعة من الفئات، والتي قد يتم إنشاء العديد منها حديثًا بواسطة مطوريها.

موضوعات ذات صلة

تمت تغطية الكائنات والفئات بمزيد من العمق في **الوحدة 4: البرمجة**.

معالجة الأحداث (Event Handling)

بمجرد إنشاء التصميم المرئي لواجهة تطبيقك وتسمية الكائنات بشكل معقول، فإن الخطوة الآتية هي وضع كود الخوارزميات التي تحسب المخرجات أو الإجراءات المطلوبة. في حالة تطبيق تحويل درجة الحرارة الخاص بنا، يعني هذا التحويل بين إدخال درجة الحرارة بالدرجات فهرنهايت إلى الإخراج بالدرجات المئوية والعكس صحيح.

يتم تنفيذ هذه الحسابات عندما يضغط المستخدم على أحد الزرين اللذين أضفناهما إلى النموذج الرئيس للتطبيق. عندما يضغط المستخدم على زر (أو ينفذ أي إجراء يتعرف إليه نظام التشغيل) فإنه يقوم بتشغيل حدث معين - يسمى في هذه الحالة "حدث عند النقر".

من أجل ربط مشغل الحدث بالعمليات الحسابية، تحتاج إلى إنشاء معالج أحداث. هناك العديد من الطرق لتحقيق ذلك في Android Studio. تتضمن الطريقة الأكثر تقليدية تسجيل طريقة مستمع (a listener method) لكل زر يقوم بتشغيل طريقة **معالج الأحداث** المشفرة بشكل خاص.

قم بإعادة زيارة قسم **مستكشف المشروع**، وقم بتوسيع (Java) ثم (MainActivity). يؤدي النقر فوق هذا إلى عرض كود برنامج Java، والذي سيتم إنشاؤه تلقائيًا لك. ستحتاج إلى إدراج الكود المميز الموضح في الشكل 7.29 لإضافة معالجات الأحداث المناسبة التي ستستجيب لمدخلات المستخدم وتحسب المخرجات المرغوبة من أجل الحصول على مكونات الشاشة الصحيحة.

المصطلح الرئيس

معالجة الأحداث – تصف عملية استخدام دالة مكتوبة خصيصًا أو طريقة من طرق الفئة لتنفيذ إجراءات محددة عند تفعيل حدث من قبل المستخدم أو النظام. على سبيل المثال، يمكن عرض نتيجة عملية حسابية عندما يتم الضغط على زر، أو عرض مؤشر "البطارية منخفضة" على الجهاز عندما ينخفض مستوى الشحن إلى حد معين.

```
import android.support.design.widget.Snackbar;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.Button;
import android.widget.EditText;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
        fab.setOnClickListener((view) -> {
                Snackbar.make(view, "Replace with your own action", Snackbar.LENGTH_LONG)
                        .setAction("Action", null).show();
        });

        addButtonCtoFListener();     //call method to add listener to CtoF button
        addButtonFtoCListener();     //call method to add listener to FtoC button
    }
```

الشكل 7.29 إضافة كود جديد إلى تطبيقك

تقوم الخطوط الجديدة (المميزة) الموضحة في الشكل 7.29 بتنفيذ إجراءين.

1. تقوم باستيراد حزم كود Java المختلفة التي نحتاجها لتطبيقنا لتجميعها في برنامجنا.
2. تطلب طريقتين للمستمع، واحدة لكل زر في نموذج التطبيق.

مهمتك الآتية هي إنشاء كود Java لطريقتي المستمع. سيتضمن كل منها معالج الأحداث الذي:

- يحصل على القيمة التي تم إدخالها في TextField في نموذج التطبيق
- يقوم بإجراء الحساب الصحيح
- يعيد النتائج إلى TextField الآخر في نموذج التطبيق.

يوضح الشكلان 7.30 و7.31 الكود الجديد لكل مستمع ومعالج. يجب ملاحظة أن كود Java لطريقة addButtonCtoFlistener مُعلَّم لإظهار أفضل الممارسات.

```
/*
 * Class method adding a Listener to
 * the CtoF button to enable responding to
 * events
 */
public void addButtonCtoFListener() {

    //Create button object representing CtoF button on form
    Button btnCtoF = (Button)findViewById(R.id.btnCtoF);

    /*
     * OnClick listener for CtoF button will
     * call its associated event handler to perform
     * the CtoF calculation
     */
    btnCtoF.setOnClickListener(new OnClickListener() {
        public void onClick(View v) {
            float celsius;       // Inputted Celsius value
            float fahrenheit;    // Calculated Fahrenheit value

            //alias Celsius input box
            EditText cValue = (EditText) findViewById(R.id.cText);

            //grab celsius value from input box, converting from text to floating point
            celsius = Float.valueOf(cValue.getText().toString());

            //perform conversion of Celsius to Fahrenheit
            fahrenheit = (celsius * 9.0f / 5.0f) + 32.0f;

            //alias Fahrenheit input box
            EditText fValue = (EditText) findViewById(R.id.fText);

            //convert calculated Fahrenheit floating point value to text
            String fahrenheitText = Float.toString(fahrenheit);

            //put the converted Fahrenheit text value into the Fahreneit input box
            fValue.setText(fahrenheitText);
        }
    });

}
```

الشكل 7.30 كود جافا لطريقة addButtonCtoFListener (مُعلَّم)

```
public void addButtonFtoCListener() {

    Button btnFtoC = (Button)findViewById(R.id.btnFtoC);

    btnFtoC.setOnClickListener(new OnClickListener() {
        public void onClick(View v) {
            float celsius;
            float fahrenheit;
            EditText fValue = (EditText) findViewById(R.id.fText);
            fahrenheit = Float.valueOf(fValue.getText().toString());
            celsius = (fahrenheit - 32.0f) * 5.0f / 9.0f;
            EditText cValue = (EditText) findViewById(R.id.cText);
            String celsiusText = Float.toString(celsius);
            cValue.setText(celsiusText);
        }
    });

}
```

الشكل 7.31 كود جافا لطريقة addButtonFtoCListener (غير مُعلَّم)

فكِّر مليًّا

اقرأ الكود المشروح لطريقة addButtonCtoFlistener وقارنها بالكود المماثل، ولكن غير المعلق، في addButtonftoClistener. هل يمكنك إضافة تعليقات مناسبة إلى طريقة addButtonftoClistener لتحسين قابليتها للقراءة؟

التعليق التوضيحي للكود

كما رأيت، من المهم التأكد من توثيق الكود الخاص بك بشكل مناسب. يعني التعليق التوضيحي للكود بشكل أساسي إدراج تعليقات يمكن للمطورين قراءتها في جميع أنحاء التعليمات البرمجية. على الرغم من إزالة هذه التعليقات في أثناء عملية التجميع وبالتالي يتعذر الوصول إليها من قبل المستخدم، إلا أن إدراجها يعد ممارسة جيدة.

الغرض من إضافة تعليقات توضيحية إلى كود البرنامج الخاص بك بتعليقات ذات مغزى هو تحسين قابلية قراءته والمساعدة في فهمه. يمكن أن يكون هذا مهمًا بشكل خاص عندما يُطلب من المبرمجين إعادة تطوير التطبيقات التي كتبها آخرون في الأصل.

يجب أن توضح تعليقاتك كيفية ارتباط التعليمات البرمجية بتطبيقها في العالم الحقيقي، وليس شرح صيغة الكلمات المحجوزة الفعلية المستخدمة. تختلف التقنيات المستخدمة للتعليق على كود البرنامج بين لغات البرنامج.

تستخدم Java مزيجًا من التعليقات متعددة الأسطر والتعليقات أحادية السطر، ويظهر ذلك في الشكل 7.32. استخدام شائع آخر للتعليق التوضيحي هو (التعليق) على كود البرنامج الذي يريد المطور الاحتفاظ به ولكن ليس تجميعه، ربما في أثناء اختبار الأفكار الجديدة أو تحديد الكود الذي لا يعمل بشكل صحيح.

المهارات

- اختيار أدوات وأنظمة تكنولوجيا المعلومات المناسبة المستخدمة لتطوير حلول تطبيقات الأجهزة المحمولة

```
1  public class HelloWorld{
2
3      /*
4          Main function
5
6          Program to test program output
7
8          Written by A. Jones
9          Version 1.0
10
11     */
12     public static void main(String []args){
13
14         // Output welcome message
15         System.out.println("Hello World");
16
17     }
18 }
```

الشكل 7.32 التعليقات متعددة الأسطر وأحادية السطر

استخدام المعرفات ذات الدلالة هو أيضًا جانب أساسي من توضيح الكود، حيث يُقال إن أسماء التمثيل التي اخترتها تقوم بتوثيق نفسها.

يمكن لبعض حزم SDK وأدوات الجهات الخارجية إنشاء وثائق صفحة الويب تلقائيًا من كود البرنامج الخاص بك إذا تم تنسيقها بالطريقة الصحيحة. على سبيل المثال، تقوم أداة Javadoc من Oracle بتنفيذ هذه الوظيفة لـ Java.

بحث

تفضل بزيارة www.oracle.com للتحقق من أداة Javadoc من Oracle، ومعرفة كيفية دمج التعليقات المتوافقة مع JavaDOC في التعليمات البرمجية وإنشاء صفحات ويب قابلة للتنقل تلقائيًا لها.

الاستفادة من إمكانات الجهاز

يدعم Android وApple مطوري تطبيقات الأجهزة المحمولة من خلال توفير **واجهات برمجة التطبيقات (APIs)** و**الأطر** التي تسمح للتطبيقات بتلقي بيانات الحركة من الأجهزة المدمجة مثل الجيروسكوبات ومقاييس التسارع. يطلق Android على هذا إطار Sensors بينما تشير Apple إلى تنفيذها باسم إطار Core Motion.

المصطلحات الرئيسة

واجهة برمجة التطبيقات (API) - تعمل كمكتبة من الإجراءات الروتينية المكتوبة مسبقًا والتي توفر للمطور إمكانية الوصول إلى أنظمة أخرى (مثل قواعد البيانات) أو نظام تشغيل الجهاز أو أجهزته الفعلية. الهدف من واجهة برمجة التطبيقات هو تسهيل عملية التطوير من خلال التخلص من (إخفاء) تعقيد أنظمة البرامج والأجهزة الموجودة تحتها.

الإطار — مجموعة معينة من أدوات التطوير التي يمكن الوصول إليها عبر واجهة برمجة التطبيقات الخاصة بها.

التحقق من حالة الجهاز

جزء أساسي من برمجة تطبيقات الأجهزة المحمولة هو القدرة على الاستفسار عن حالة الجهاز. قد يقوم المطور بفحص أنواع مختلفة من الحالات، بما في ذلك الإحصائيات المهمة مثل الموقع الفعلي أو مستوى البطارية أو الاتجاه (أي ما إذا كان الوضع عموديًا أو أفقيًا). على سبيل المثال، يستخدم Android فئة BatteryManager المفيدة، والتي توفر طريقة للاستعلام عن خصائص البطارية والشحن.

على الرغم من أنك لست بحاجة إلى إجراء هذا النوع من الاستفسار في تطبيق تحويل درجة الحرارة التمهيدي، إلا أن القدرة على القيام بذلك تعد مهارة مهمة.

بحث

يمكن العثور على العديد من الأمثلة التي توفر رمز Java الضروري لتعرف حالة جهاز Android. قم بزيارة موقع www.tutorialforandroid.com الذي يوضح كيف يمكن التحقق من مستوى بطارية هذا الجهاز.

تحقق من هذا الكود وقم بتكييفه ودمجه في مشاريعك الخاصة.

اتجاه الجهاز

يحتاج تطبيقك إلى الاكتشاف التلقائي لاتجاه الجهاز الفعلي الذي يعمل عليه (ومتى يتغير). يقوم بذلك عن طريق الاستعلام عن دوران الشاشة. يُستمد الدوران العادي للجهاز من شكل الشاشة: على سبيل المثال، إذا كانت الشاشة طويلة بشكل طبيعي (عمودي) وتم تدوير الجهاز جانبيًا (أفقيًا)، فستكون قيمة دورانه إما 90 أو 270 درجة اعتمادًا على الاتجاه الذي تم تشغيله فيه (في اتجاه عقارب الساعة) وإما عكس اتجاه عقارب الساعة). بهذا توفر أندرويد فئة مناسبة تُدعى "Display" ضمن واجهة برمجة التطبيقات الخاصة بها لمساعدة المطورين في تحديد دوران الشاشة وحجمها ومعدل التحديث.

يمكن للتطبيقات أيضًا فرض وضع التوجيه عن طريق تغيير دوران الجهاز تلقائيًا ليناسب العرض الرأسي أو الأفقي، على الرغم من أن هذا لا يعتبر عمومًا ممارسة جيدة.

بحث

تحقق من فئة (العرض) في Android وأساليبها من خلال زيارة https://developer.android.com

إنشاء ملف تنفيذي لجهاز مستهدف

يمكنك نشر تطبيقك إما على جهاز تمت محاكاته وإما الجهاز الفعلي الذي استهدفته.

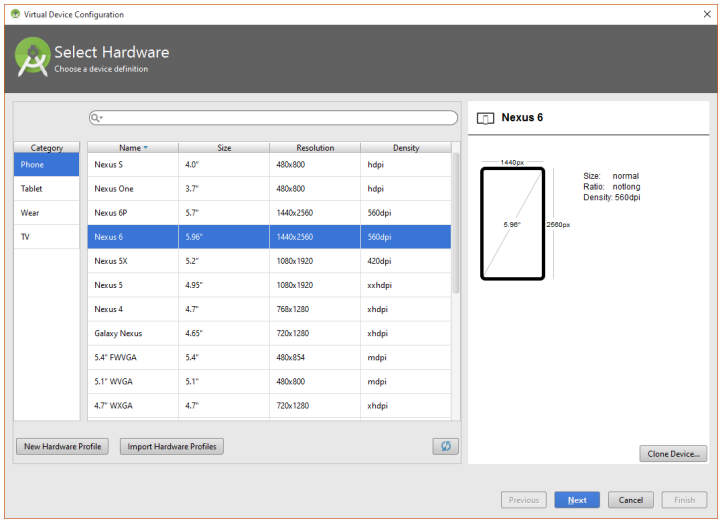
جهاز تمت محاكاته

يعد استخدام جهاز تمت محاكاته مفيدًا لأنه يتيح لك اختبار توافق وأداء تطبيقك على أنواع الأجهزة التي قد لا تتمكن من الوصول الفعلي إليها، على سبيل المثال الهواتف الذكية والأجهزة اللوحية وأجهزة التلفزيون المختلفة والأجهزة التكنولوجية القابلة للارتداء. في Android Studio، يمكنك تحقيق ذلك من خلال إنشاء **جهاز Android افتراضي (AVD)**. لإنشاء جهاز Android افتراضي (AVD)، من الضروري تحديد الأجهزة وإصدار نظام التشغيل Android المستخدم (X KitKat.4 أو X Lollipop.5 أو 6.0 Marshmallow).

يتيح لك Android Studio إنشاء ملفات جهاز Android افتراضي (AVD) متعددة ثم اختيار الجهاز الذي تريد استخدامه عند تطوير تطبيقك. يتيح لك ذلك اختبار تطبيقك بسهولة على العديد من الأجهزة المادية المختلفة. يمكن اختيار الجهاز من خلال تعريفات الأجهزة المتاحة، كما هو موضح في الشكل 7.33.

المصطلح الرئيس

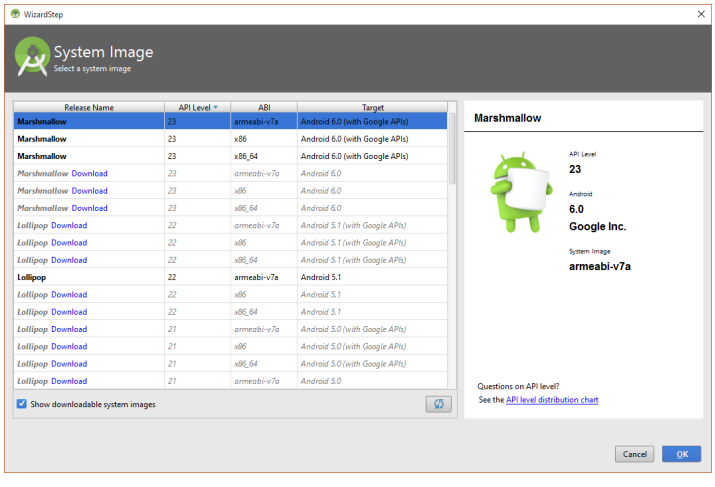
جهاز Android الافتراضي (AVD) - إصدار تمت محاكاته ببرنامج لجهاز مادي معين، يعمل داخل نظام تشغيل مضيف مثل Microsoft Windows أو Apple OS X.



الشكل 7.33 تحديد تعريف الجهاز لجهاز Android حقيقي

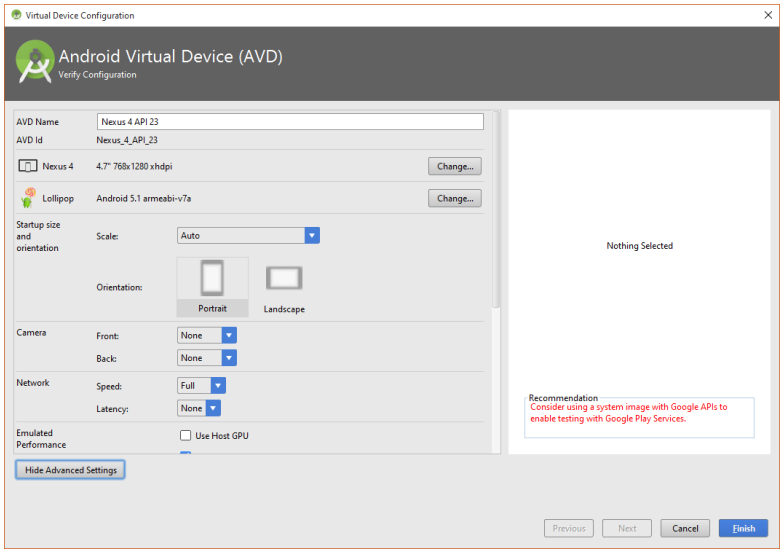
من الممكن بعد ذلك تحديد شكل الجهاز المستخدم Android لاستخدامها. وسيتم تثبيت بعض الصور في Android Studio بالفعل ولكن يمكنك تنزيل صور أخرى بحسب الحاجة (كما هو موضح في الشكل 7.34). لتشغيل تطبيقك بنجاح باستخدام جهاز Android افتراضي (AVD)، يجب عليك استخدام صورة نظام متوافقة مع ميزات البرمجة أو الأجهزة التي استخدمتها.

تتوفر كل صورة من صور النظام عادةً في خيارات المحاكاة الافتراضية المختلفة: x86 (32 بت) أو x64 (64 بت) أو ARM (آلة حوسبة مجموعة التعليمات المخفضة المتقدمة). تعد آلة ARM بشكل عام الأبطأ ولكنها الأكثر استقلالاً عن خيارات المحاكاة الافتراضية المحددة في BIOS جهاز الحاسوب المضيف (نظام إخراج الإدخال الأساسي)؛ ويجب دائمًا تعديل هذه الخيارات بحذر.



الشكل 7.34 اختيار صورة لنظام أندرويد لجهاز Android افتراضي (AVD)

من الممكن بعد ذلك تخصيص خيارات متنوعة لتفضيلاتك الشخصية، كما هو موضح في الشكل 7.35.



الشكل 7.35 إعداد تفضيلات جهاز Android الافتراضي (AVD) الخاص بك

الشكل 7.36 AVD يعمل على نظام تشغيل Microsoft Windows 10

يوضح الشكل 7.36 جهاز Android افتراضي (AVD) قيد التشغيل في نظام التشغيل Microsoft Windows 10.

الشكل 7.37 جهاز Android متصل عبر كبل USB

جهاز مادي

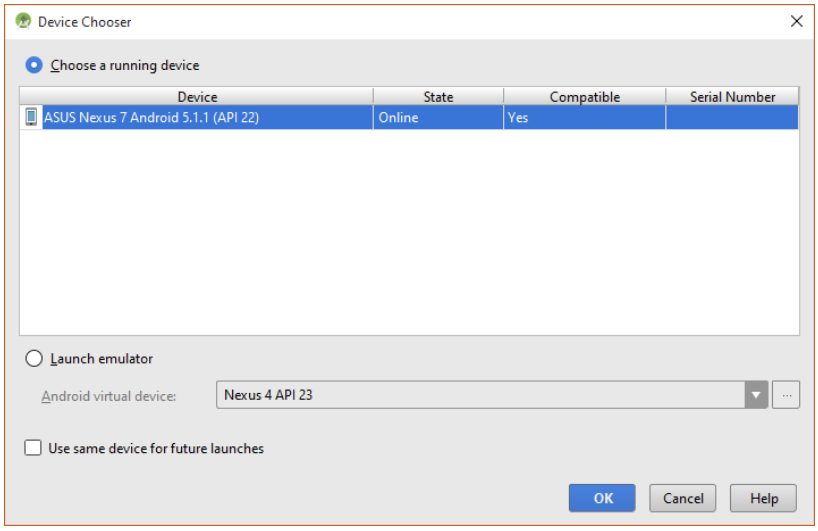
تتمثل التقنية الأخرى لاختبار تطبيق Android الخاص بك في توصيل النظام الذي تقوم بتطويره عليه (على سبيل المثال Apple Mac أو Microsoft Windows PC) بهاتف ذكي أو جهاز لوحي أو تقنية يمكن ارتداؤها تعمل بنظام Android عبر كبل USB متوافق، كما هو موضح في الشكل 7.37.

بالإضافة إلى تثبيت برنامج تشغيل Android Debug Bridge (ADB) الصحيح للنظام المضيف الخاص بك للتحدث إلى جهاز Android، غالبًا ما يكون من الضروري تعديل بعض إعدادات الجهاز لتشغيل تطبيقك. يتضمن هذا عادةً تمكين خيارات USB أو تشغيل التطبيقات التي لم يتم تنزيلها من خلال Google Play، ويجب إجراؤها مرة واحدة فقط. يمكن أن تختلف إعدادات المطور هذه من جهاز لآخر، وقد يتم إخفاؤها عمدًا لمنع التلف العرضي، لذلك يُنصح بالتحقق من الإجراء الصحيح مع الشركة المصنعة للجهاز قبل محاولة ذلك.

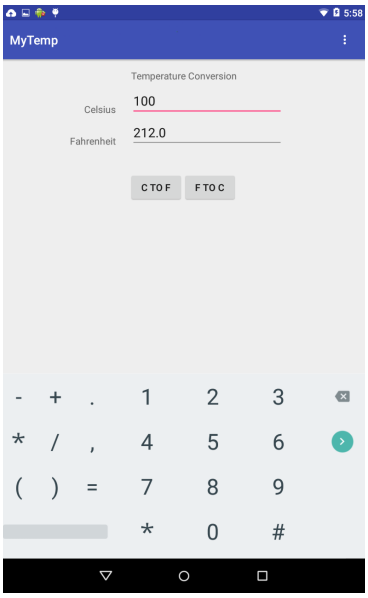
الميزة الأساسية لتشغيل تطبيقك على الجهاز الفعلي المستهدف هي أنك ستحصل على الانطباع الحقيقي عن أدائه وسلوكه. تحتوي العديد من الأجهزة أيضًا على أدوات تصحيح المطور التي يمكن أن تساعد في الاختبار المباشر.

بصفتك مطورًا، تحتاج فقط إلى تحديد جهاز Android المتصل (وليس الافتراضي) بمجرد النقر فوق زر (تشغيل) التطبيق الأخضر في Android Studio. إذا تم توصيل الجهاز بنجاح، فيجب إدراجه كجهاز قيد التشغيل، والذي يمكن اختياره بعد ذلك. بمجرد تحديده والنقر على زر **موافق**، سيبدأ Android Studio بعد ذلك في نقل كود Java الضروري إلى الجهاز عبر USB وبدء تشغيل التطبيق؛ العملية سريعة نسبيًا.

يوضح الشكل 7.38 أنه تم اختيار جهاز فعلي كمنصة مستهدفة ويوضح الشكل 7.39 تطبيق تحويل درجة الحرارة الذي تم نقله وتشغيله على جهاز فعلي.



الشكل 7.38 تحديد جهاز فعلي كمنصة مستهدفة



الشكل 7.39 تم نقل تطبيقك وتشغيله على جهاز فعلي

تظهر لوحة المفاتيح الافتراضية للجهاز الفعلي تلقائيًا في أثناء تشغيل التطبيق، ما يمنع المستخدم من إدخال الأرقام فقط. يحدث هذا بسبب قرار التصميم باختيار حقول النص (الرقم (الموقّع)).

على الرغم من أن الاختبار السريع يبدو أنه يشير إلى أن التطبيق يعمل بشكل صحيح (100 درجة مئوية تعادل بالفعل 212 درجة فهرنهايت)، يجب أن تكون خطوتك التالية هي مراجعة جهودك من خلال عملية تسمى مراقبة الجودة.

المهارات

- مهارات التحليل واتخاذ القرار
- القدرة على العمل بطريقة قانونية وأخلاقية

مراقبة الجودة

مراقبة الجودة هي عملية تستعرض معايير التصنيع المطبقة في أثناء الإنتاج. هدفها هو تقديم أفضل منتج أو خدمة للعميل. بالنسبة لتطبيق الجهاز المحمول، يعني هذا بشكل أساسي قياس مدى جودة تنفيذ التصميم في أثناء التطوير للجهاز (الأجهزة) المستهدفة.

تعد مراقبة الجودة جزءًا لا يتجزأ من عملية تطوير تطبيقات الأجهزة المحمولة، مما يساعدك على تقييم المعيار الناتج الذي حققته عملية التطوير الخاصة بك، والأهم من ذلك، تحديد مجالات التحسين المحتملة. يجب إجراء مراقبة الجودة قبل بدء الاختبار الرسمي. يتم عرض نظرة عامة على مراقبة الجودة في الجدول 7.7.

الجدول 7.7 مجالات الاهتمام بمراقبة الجودة والأسئلة والاعتبارات الرئيسة

الاعتبار	السؤال	قم بمراعاة...
الكفاءة والأداء (وهذا ما يسمى تطبيق (ملف التعريف))	كيف يؤثر تطبيقك في موارد الجهاز المحمول؟ هل يمكننا تحديد أي مشاكل تؤثر على الأداء؟	• استخدام تخصيص ذاكرة الوصول العشوائي • استخدام وسائط تخزين الجهاز • استخدام وحدة المعالجة المركزية (CPU)، لا سيما من حيث حصتها المئوية • استخدام عمليات النظام والتطبيق التي قد تسلط الضوء على اختناقات الأداء • استخدام وحدة المعالجة الرسومية (GPU)، خاصة عند عرض عناصر واجهة المستخدم (UI) • استخدام البطارية، مقارنة استنزاف التطبيقات المماثلة • قيود درجة الحرارة - يمكن إيقاف تشغيل الأجهزة المحمولة لحماية المكونات في أثناء فترات الاستخدام المكثف (خاصة عند إعادة الشحن) إذا أصبحت ساخنة جدًا
قابلية الصيانة	ما مدى سهولة تعديل تطبيقك أو تحسينه؟ هل قمت ببرمجة التطبيق بشكل مناسب؟	• معايير كتابة الكود الموصى بها • مستويات وفائدة التعليق التوضيحي للكود • هيكل وتنظيم كود البرنامج الخاص بك • قابلية التوسع في الحل الخاص بك • استخدام بنيات البرمجة والدالات والإجراءات وأنواع البيانات والفئات وإطار كود الجهاز
قابلية النقل	ما الأجهزة المتوافقة مع تطبيقك؟ كيف يمكنك تحسين التوافق المحتمل؟	• الأجهزة التي ستقوم بتشغيل تطبيقك بشكل صحيح • الأجهزة التي لا تقوم بتشغيل تطبيقك (وتحدد السبب (الأسباب)) • الأجهزة التي تشغل تطبيقك ولكن بها مشكلات بسيطة (وما الذي يمكن تعديله لتحسين قابلية النقل) • أي مشكلات توافق غير متوقعة لم تكن تتوقعها • ما إذا كانت قابلية النقل الفعلية لتطبيقك تلبي الأنظمة الأساسية المستهدفة في مرحلة التصميم (بما في ذلك إصدارات محددة من نظام التشغيل).
إمكانية الاستخدام	ما مدى سهولة تفاعل المستخدم مع تطبيقك؟ كيف يمكنك تحسين تجربة المستخدم؟	• الالتزام بإرشادات الشركة المصنعة، مثل مبادئ ومعايير التصميم الموصى بها من Apple iOS وAndroid • التقاط تعليقات المستخدمين، سواء كانت جيدة أو سيئة • تحديد إبداءات الإعجاب وعدم الإعجاب الشائعة • طلب ملاحظات المستخدم للتحسين • تحديد ما إذا كان التطبيق يلبي متطلبات المستخدم المستهدفة في مرحلة التصميم.

موضوعات ذات صلة

لمزيد من المعلومات حول ذاكرة الوصول العشوائي ووحدة المعالجة المركزية ووحدة معالجة الرسوم، راجع **الوحدة 8: تطوير ألعاب الحاسوب**.

للتذكير بما هو المقصود بـ (واجهة المستخدم (UI))، راجع *واجهة المستخدم*، صفحة 9.

من الأفضل إجراء مراقبة الجودة من خلال المراجعة اليدوية (بدلاً من الآلية) ويمكن أن تكون عملية تقييم ذاتي. سيفحص الاختبار الرسمي التعليقات الواردة من مجموعة واسعة من المستخدمين.

فكِّر مليًّا

يجب أن تساعدك عملية مراقبة الجودة على تقييم مستوى تطبيقك، سواء من حيث تصميمه أو تنفيذه. عند القيام بذلك، يجب أن يوفر رؤية حيوية لمستويات الأداء الخاصة بك، من خلال تسليط الضوء على المجالات المستهدفة التي تم تحقيقها وتلك التي تحتاج إلى بعض التحسين.

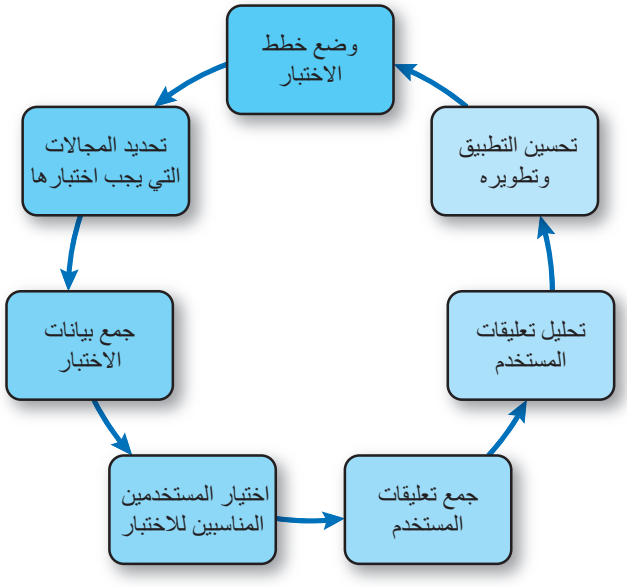
تعد مراقبة الجودة عن طريق تقييم الذات مكونًا رئيسًا في أي عملية لتحسين الذات وهي ضرورية للترقي وفي النهاية، لتحسين الأداء في مكان العمل. فكر في تصميم وتنفيذ تطوير تطبيق تحويل درجة الحرارة. كيف يمكن تحسينه؟

اختبار تطبيق جهاز محمول

يُعترف بالاختبار في نطاق واسع باعتباره عملية أساسية تضمن أن أي تطبيق هاتف محمول تقوم بتطويره يلبي المتطلبات التي تم تحديدها في أثناء تصميمه، وأنه يعمل بطريقة دقيقة وموثوقة وقوية. يجب إجراء الاختبار في كل مرة يتم فيها تحديث التطبيق، لذلك يجب أن يُنظر إلى العملية على أنها دورية؛ يظهر هذا في الشكل 7.40.

المهارات

- مهارات التحليل واتخاذ القرار
- مهارات الإدارة الذاتية والتخطيط
- القدرة على العمل بطريقة قانونية وأخلاقية



الشكل 7.40 المراحل الدورية التي ينطوي عليها الاختبار القوي لتطبيق الجهاز المحمول

يجب أن توضح خطط الاختبار خطوة بخطوة كيف تخطط لاختبار تطبيق الهاتف المحمول الخاص بك. سيشمل ذلك تحديد العديد من (حالات الاستخدام). تحاول كل حالة استخدام سرد قصة تفاعل المستخدم مع تطبيقك (الناجح وغير الناجح) وعادةً ما تتضمن مزيجًا من:

- بيانات الاختبار التي يتم إدخالها
- العمليات التي يتم تنفيذها
- ترتيب العمليات التي يتم تنفيذها
- ردود المستخدم على مطالبات التطبيق الخاص بك.

يمكن العثور على بيانات الاختبار عادةً في واحدة من ثلاث حالات محتملة.

- **عادي** - البيانات ضمن النطاق المقبول الذي يدخله المستخدم عادةً.
- **طرفي** - بيانات غير محتملة عند حواف النطاق المقبول.
- **خاطئ** - البيانات التي لا يجب إدخالها (مثل النص بدلاً من الأرقام).

هناك مفهوم آخر قد تواجهه كجزء من خطة الاختبار وهو فكرة اختبار الصندوق الأبيض والأسود. عادةً ما يتم إجراء اختبار الصندوق الأبيض بواسطة المطور من خلال تتبع حالات الاستخدام من خلال منطق رمز البرنامج وإكمال **جداول التتبع**. يتم إجراء اختبار الصندوق الأسود من قبل المستخدم باتباع حالة الاستخدام. لا يتعرض المستخدم لرمز البرنامج ولا يحتاج إلى معرفة كيفية عمل التطبيق، بينما أنت، كمطور، مهتم فقط بالنتيجة التي يحصل عليها المستخدم.

تشمل المجالات التي تحتاج إلى اختبار ما يلي:

- **الوظيفة** - يجب أن تعمل جميع الوظائف في التطبيق كما هو متوقع.
- **القبول** - يجب أن يكون التطبيق مناسبًا للغرض: أي أنه يجب أن يلبي متطلبات المستخدم المحددة.
- **التثبيت** - يغطي التثبيت الأولي للتطبيق (التنزيل من متجر التطبيقات) وتحديثه وإزالته في نهاية المطاف.
- **الأداء** - مدى جودة عمل التطبيق وتأثيره على الجهاز (الأجهزة) المستهدف.
- **سهولة الاستخدام** - يجب أن يكون التطبيق سهل الاستخدام وأن يتقن مستخدموه واجهته بسرعة؛ غالبًا ما يكون الحصول على استجابة إيجابية من المختبرين جزءًا أساسيًا من نجاح التطبيق.
- **التوافق** - يجب أن يعمل التطبيق بشكل جيد باستمرار عبر النماذج والعلامات التجارية المختلفة للأجهزة.

المصطلح الرئيس

جدول التتبع - جدول يتم إنشاؤه بواسطة المطور من خلال اختبار الصندوق الأبيض، والذي يرسم التغييرات في متغيرات البرنامج عند تنفيذ حالة الاستخدام. سيكون لجدول التتبع إدخالات لكل حالة استخدام مع النتائج المتوقعة (ما يجب أن يحدث) والنتائج الفعلية (ما يحدث بالفعل). يساعد مقارنة هذه النتائج المطور على تقييم ما إذا كان التطبيق يعمل بشكل صحيح.

اختبار المستخدمين وملاحظات المستخدم

يُنصح باختيار مستخدمي الاختبار الذين ليس لديهم علاقة تذكر بعملية التطوير. سيضمن ذلك التعبير عن ملاحظاتهم بصدق ودون تحيز. يجب أن يتم اختيار المستخدمين من عينة سكانية عشوائية، مثاليًا، ذات مستويات مختلطة من الخبرة التقنية.

يجب جمع التعليقات من المستخدم. يمكن تحقيق ذلك باستخدام مجموعة متنوعة من تقنيات تقصي الحقائق بما في ذلك:

- الملاحظة المباشرة
- المقابلات الفردية
- مجموعات التركيز
- الاستبيانات أو الاستطلاعات
- تقارير الأخطاء الآلية التي تم إنشاؤها بواسطة التطبيق نفسه.

يمكن أن يكون جمع تعليقات المستخدمين مستهلكًا للوقت ومكلفًا، لذلك يجب استخدام التقنيات بحكمة. من الأساليب الشائعة (والفعالة من حيث التكلفة) تسجيل التعليقات من المستخدمين المؤكدين الذين يستخدمون متاجر التطبيقات عبر الإنترنت لأنها مجانية وغير مرغوب فيها (على سبيل المثال من متجر تطبيقات Apple أو Google Play). تعد إطلاق إصدارات Alpha أو Beta المجانية (المبكر جدًا) من تطبيق للمستخدمين المحددين أيضًا تكتيكًا مفيدًا يجب مراعاته.

من الناحية المثالية، ينبغي أن تكون الملاحظات عبارة عن مزيج من البيانات الكمية والنوعية التي يمكن تجميعها وتحليلها لإنتاج معلومات مفيدة وتحديد الأنماط والاتجاهات في تجربة المستخدم، والتي يمكن أن تسلط الضوء على العيوب ومجالات التحسين.

البيانات الكمية - البيانات التي تحدد الكميات: أي أنه يمكن قياسها وكتابتها كأرقام أو نسب مئوية أو نسب. على سبيل المثال، (50 في المائة من المستخدمين لم يواجهوا أي مشاكل في أثناء استخدام التطبيق).

البيانات النوعية - يتم تمثيلها بالأسباب والآراء والدوافع ويمكن استخدامها لاستكشاف (وفهم) البيانات الكمية. على سبيل المثال، (وجد المستخدمون صعوبة في التنقل في التطبيق وصعوبة قراءة النص على الأجهزة الصغيرة).

مناقشة

انظر إلى مثالين للنتائج الكمية والنوعية في الصفحة 40. هل يمكن العثور على رابط؟ هل يُفسر سبب تعرض 50 في المئة من المستخدمين لمشاكل عند استخدام التطبيق (نتيجة كمية) بالمشكلات التي تمت مواجهتها في التنقل وحجم النص (نتيجة نوعية)؟ ناقش في مجموعات صغيرة.

يوضح الجدول 7.8 الأسئلة الكمية والنوعية في الاستبيان.

الجدول 7.8 الأسئلة الكمية والنوعية

السؤال الكمي 10 أ	كيف كانت واجهة مستخدم MyApp؟	☐ ضعيفة	☐ مقبولة	☐ جيدة	☐ ممتازة
السؤال النوعي 10 ب	يرجى توضيح أسبابك...				

على سبيل المثال، إذا أشارت تعليقات المستخدمين إلى أنه في حين وجد البعض واجهة المستخدم (جذابة وذات نظام ألوان ساطع) (بيانات نوعية)، ستجد أيضًا أن (60 في المائة من المستخدمين اعتبروا واجهة المستخدم ضعيفة ((بيانات كمية)، فمن الواضح أن هناك مشكلة في قابلية الاستخدام في تصميم التطبيق.

إذا كانت البيانات الداعمة غير متوافرة، فقد يكون من الضروري العودة إلى هؤلاء المستخدمين والتحقيق في ردودهم غير الراضية للعثور على السبب الأساسي. يجب أن يسأل نهج تقصي الحقائق المتميز دائمًا عن الأسباب وليس مجرد تسجيل نتيجة.

تحسين تطبيقك

يتم توجيه عملية تحسين تطبيقك ومراجعته من خلال تحليل النتائج المجمعة للاختبار (أي من قبل المطور) وتعليقات المستخدم المستلمة.

يجب أن تشمل القضايا التي يجب معالجتها ما يلي:

- التأكد من أن التطبيق يحتوي على جميع الوظائف المتفق عليها الموضحة في التصميم
- إصلاح أخطاء وقت التشغيل أو الأخطاء أو أعطال التطبيقات لتحسين الموثوقية
- إصلاح الحسابات والمخرجات غير الدقيقة
- تحسين الاستجابة
- تحسين واجهة المستخدم (على سبيل المثال، نظام الألوان، الخط، أحجام الخطوط، الصور)
- تحسين التنقل والحركة داخل التطبيق
- زيادة السرعة/الأداء
- تقليل استخدام التطبيق لموارد الجهاز من خلال زيادة تحسين الأصول أو ببساطة البرمجة بشكل أكثر كفاءة
- تحسين توافق التطبيق مع الموديلات/العلامات التجارية المختلفة للأجهزة المحمولة.

فكِّر مليًّا

ستكون قد راجعت تطبيقك باستخدام تعليقات من زملائك المحترفين ومستخدمي الاختبار. غالبًا ما تكون الآراء التي يعبر عنها الآخرون مفيدة جدًا لأنها عادةً ما تكون غير متحيزة، مما يوفر حكمًا محايدًا حول تصميم تطبيقك وتنفيذه. تعلم كيفية الاستجابة للنتائج والاستفادة من التعليقات في مسيرتك. سيمكنك هذا من التفاعل بطريقة ناضجة تؤدي إلى تحسينات في كل من المنتج النهائي وقدراتك على حل المشكلات.

يجب تسجيل أي تغييرات يتم إجراؤها على التطبيق في سجل التغيير، وهو مستند مكتوب من قبل المطور يوضح بالتفصيل الإصلاحات والتحسينات التي تم إجراؤها لكل إصدار من التطبيق. يتم استخدام رقم الإصدار لتحديد كل إصدار متتالي بوضوح. يستخدم هذا نظام (major.minor)، على سبيل المثال:

- MyApp الإصدار 1.0 - أول إصدار رئيس.
- myApp ver 1.1 - تحديث بسيط، قائمة الإصلاح (من تعليقات المستخدمين)، تحسين التوافق.
- myApp ver 2.0 - تحديث رئيس، يتضمن وظائف إضافية.

كما ترى، تزداد الإصدارات الثانوية عند إجراء تحديثات وإصلاحات صغيرة (على سبيل المثال، myApp ver 1.1). قد تخرج العديد من الإصدارات الثانوية بين التحديثات الرئيسة. في المقابل، تمثل التحديثات الرئيسة تغييرات كبيرة في وظائف التطبيق أو ميزاته أو رمزه (على سبيل المثال، myApp ver 2.0). قد تكون لديك أيضًا تجربة شخصية مع إصدارات التطبيقات اللاحقة التي تعرض أخطاء وأعطال جديدة تمامًا. هذا أمر شائع جدًا وخطر يواجهه كل مطور عند محاولة معالجة مشكلة محددة؛ في إصلاح الخلل، يقدمون أحيانًا الآخرين.

الدروس المستفادة من تطوير تطبيق جهاز محمول

من المهم أن تعكس وتقيم فعالية التطبيق الذي قمت بتطويره. تشجعك هذه المراجعة على طرح الأسئلة التالية على نفسك:

- ما مدى تلبية الحل لمتطلبات المستخدم المحددة في وثيقة التصميم؟

فكر في: كيف استوفى تطبيقك متطلبات المستخدم المستهدف، وما العناصر التي حققها والعناصر المفقودة أو غير المكتملة؟

- ما المشكلات التي نشأت في أثناء الاختبار ومراجعتك للتطبيق؟

فكر في: المشكلات التي كان يجب أن تتوقعها، وكيف كان بإمكانك منع حدوث هذه المشكلات، وتقنيات كتابة الكود أو ميزات الجهاز التي كان من الممكن استخدامها بشكل أفضل.

- كيف يمكن تحسين التطبيق لتلبية متطلبات المستخدم عن كثب؟

فكر في: الاختبار وملاحظات المستخدم والجوانب التي يمكن إضافتها أو مراجعتها.

- مع الإدراك المتأخر الذي اكتسبته من خلال الاختبار وتعليقات المستخدمين، إذا كنت ستكرر المهمة، فهل كانت هناك حلول بديلة كان ينبغي تنفيذها بدلاً من ذلك؟

فكر في: الاختبار وتعليقات المستخدمين والتصميمات البديلة في مستند التصميم الخاص بك.

مهما كانت النتيجة، هناك دائمًا مجال للتحسين عند تطوير تطبيق جهاز محمول. أحد الأجزاء الرئيسة لعملية التعلم (وتطوير تطبيقات أفضل) هو التعلم من أخطائك.

مراجعة المهارات والمعارف والسلوكيات

عند تنفيذ تطبيقات الأجهزة المحمولة، ستحتاج إلى إظهار مهارات ومعرفة وسلوكيات معينة، بما في ذلك:

- مهارات التخطيط والتسجيل
- تحديد الأهداف ذات الصلة مع الجداول الزمنية
- التخطيط لكيفية ووقت جمع التعليقات من الآخرين
- مراجعة التعليقات من الآخرين والرد عليها
- المهنية، وآداب السلوك، ودعم الآخرين، والقيادة المناسبة في الوقت المناسب، والمساءلة
- تقييم الأهداف للحصول على رؤى حول أدائك.

فكِّر مليًّا

يحتاج مطورو تطبيقات الأجهزة المحمولة إلى تطوير وإظهار السلوكيات التي تناسب الطبيعة المهنية وتوقعات الصناعة. بعض السمات الرئيسة التي يجب عليك اعتمادها هي:

- احترام الآخرين، وخاصة آرائهم عندما يقدمون لك ملاحظات أساسية
- التواصل بشكل مناسب مع الجمهور المستهدف، وتقديم وقبول الأفكار والآراء المفتوحة والصادقة
- امتلاك عقل متفتح للأفكار الجديدة والحلول البديلة وعدم الخوف من تجربة تقنيات جديدة لتحسين الحل أو تجربة المستخدم المستهدف
- الالتزام بالتحسين الشخصي، من خلال تطوير المعرفة الجديدة والمهارات العملية
- الرغبة في تحقيق التميز من خلال معايير حل المشكلات المطبقة وتطبيقات الأجهزة المحمولة التي تم إنشاؤها.

وقفة للتفكير هل يمكنك شرح نتاج التعلم؟ ما العناصر التي وجدتها أسهل؟

تلميح وصف الخطوات العامة لتطوير تطبيق من تصميم.

توسيع الأفق ما الفرق بين مراقبة الجودة والاختبار الرسمي؟

فكِّر مليًّا

بمجرد اكتمال المشروع بالكامل، يجب عليك تقييم العملية بأكملها وتحديد أين يمكن إجراء تحسينات على التحقيق أو التصميم أو التنفيذ أو الاختبار.

المهارات

- مهارات التحليل واتخاذ القرار
- التواصل الكتابي الرسمي
- اختيار أدوات وأنظمة تكنولوجيا المعلومات المناسبة المستخدمة لتطوير حلول تطبيقات الأجهزة المحمولة
- مهارات الإدارة الذاتية والتخطيط
- القدرة على العمل بطريقة قانونية وأخلاقية

تمرين تقييمي 7.2 B.P3, B.P4, B.M2, BC.D2, C.P5, C.P6, C.P7, C.M3, BC.D3

لقد طلب منك المدير المباشر قيادة مشروع تطوير لتصميم وتنفيذ تطبيق هاتف محمول جديد يولد عروض أسعار لبستاني مساحات خضراء. يتضمن ذلك اختيار أسعار الخدمات المعنية (مثل البستنة والرصف والغرس والقص) وتكاليف السلع (مثل البذور والنباتات والشجيرات) التي سيتم تضمينها. يجب تحديد أسعار الخدمات المختلفة من الشركات المحلية على الرغم من أنه يجب أن يكون من السهل تكوينها في التطبيق نفسه. ومن المتوقع أيضًا وجود معدل قياسي لضريبة القيمة المضافة.

بهدف عرض أفضل الممارسات لمطور مبتدئ تم تعيينه حديثًا، ابدأ بإنتاج تصميمات لتطبيق هاتف محمول يلبي المتطلبات التي حددتها. ضمن جزء من عملية التصميم الخاصة بك، يجب عليك مراجعة التصميم مع أقرانك لتحديد فرص التحسينات. قبل البدء في التطوير، يجب عليك تبرير وتقييم قرارات التصميم الخاصة بك لزميلك المبتدئ من خلال شرح كيفية تلبيتها للمتطلبات المحددة.

بمجرد الموافقة على التصميم، يجب عليك إنتاج تطبيق الجهاز المحمول من تصميمك واختباره من حيث كفاءة العمل وسهولة الاستخدام والاستقرار والأداء. يجب بعد ذلك مراجعة التطبيق العامل للتأكد من أنه يلبي المتطلبات المحددة ويتم تحسينه بحسب الضرورة.

يجب إجراء تقييم نهائي لتطبيق الجهاز المحمول المحسّن ومقارنته بمتطلبات العميل.

التخطيط

- ما المشكلة التي يُطلب مني حلها؟ هل أفهم احتياجات العميل؟
- ما مدى الثقة التي أشعر بها في قدراتي على إكمال هذه المهمة، من حيث التصميم والترميز؟
- هل هناك أي مجالات فنية أعتقد أنني قد أعاني منها؟ إذا كان الأمر كذلك، فهل أعرف كيفية البحث عنها؟

تمرين تقييمي 7.2 *متابعة*

التنفيذ

- أعرف ما أفعله وما أريد تحقيقه.
- يمكنني التعرف من خلال التعليقات والمراجعة على أخطائي وتعديل تفكيري أو نهجي لإعادة نفسي إلى المسار الصحيح.
- أنا على استعداد لتحسين الحل الخاص بي بناءً على المتطلبات والتعقيبات المحددة.

المراجعة

- يمكنني شرح ماهية المهمة وكيف تعاملت مع المهمة، مع تبرير قراراتي في جميع المراحل.
- أستطيع أن أشرح كيف سأتعامل مع العناصر الصعبة بشكل مختلف في المرة القادمة (أي ما سأفعله بشكل مختلف).
- سأستخدم هذا كفرصة لتجربة تقنيات جديدة وتحسين حل المشكلات.

فكِّر في المستقبل

راما

مطورة صغيرة في فريق تطوير تطبيقات الأجهزة المحمولة

أعمل على تطبيقات الأجهزة المحمولة منذ حوالي عامين، وخلال هذا الوقت شاركت في تطوير العديد من التطبيقات المختلفة، بعضها لنظام Android والبعض الآخر لأجهزة Apple. على الرغم من اختلاف التقنيات واللغات تمامًا، إلا أن مبادئ التصميم متشابهة جدًا بشكل عام. على الرغم من أن تصميمي المفضل في بعض الأحيان يكون مثاليًا، إلا أن هذا ليس هو الحال في كثير من الأحيان ويجب أن أراجع التصميم مع مدخلات كبيرة من زملائي. يجب أن أكون ناضجة بشأن هذا الأمر وأن أتلقى الاقتراحات والتعليقات بطريقة إيجابية على الرغم من أن ذلك غالبًا ما يعني العودة إلى مرحلة التخطيط وتجربة أفكار أخرى.

بالطبع، في بعض الأحيان لا أتفق مع زملائي ويجب أن أبرر القرارات التي اتخذتها في أثناء عملية التصميم. طالما يمكنني إثبات أن أفكاري تلبي المتطلبات المحددة، فلا بأس بذلك عادةً. في الأساس، أظهر إبداعي وأتحمل مسؤولية فردية عن جهودي، وهو أمر يشجعه زملائي علنًا.

تركيز مهاراتك

مراجعة تصميمات تطبيقات الأجهزة المحمولة

من المهم أن تكون قادرًا على مراجعة تصميمات تطبيقات الأجهزة المحمولة، سواء بشكل فردي أو مع الآخرين، للتأكد من أنها تلبي المتطلبات المحددة. إليك بعض الأسئلة لتطرحها على نفسك لمساعدتك في القيام بذلك:

- ما الآثار المترتبة على تنفيذ تصميم غير مدروس جيدًا؟
- ما المتطلبات التي يجب تحديدها ومعالجتها من خلال التصميم الخاص بك؟
- كيف يمكنك تبرير القرارات التي اتخذتها في أثناء عملية التصميم؟
- كيف يمكنك تسجيل التعليقات التي تتلقاها من الآخرين وكيف يمكنك استخدامها لتحسين التصميم؟
- أخيرًا، بالنسبة لكل متطلب قمت بتحديده، حدد كيف يمكنك التأكد من التعامل معه في التصميم والمراجعة.

تصميم تطبيق جهاز محمول ـ أسئلة بشأن متطلبات التصميم

تمت مراجعة تصميم تطبيق الجهاز المحمول الخاص بك من قبل الآخرين. ماذا سيحدث بعد ذلك؟

- إذا تم تلقي تصميم تطبيق الجهاز المحمول بشكل سيئ وتم تحديد أوجه قصور خطيرة في فهمك للمشكلة و/أو تصميمك، فستحتاج إلى إعادة النظر في متطلبات العميل الأصلية.
- إذا كان تصميم تطبيق الجهاز المحمول مقبولًا بشكل عام ولكن تم تحديد مشكلات تتعلق بعناصر تصميمك التي تعتمد على أجهزة أو منصات معينة أو وجود أجهزة مدمجة، فعليك العمل على تحسين أفكارك.
- إذا كان تصميم تطبيق الجهاز المحمول مقبولًا جدًا، ففكر في أنه ما يزال هناك مجال للتحسين واطلب المشورة والتوجيه من الآخرين الذين قد يكون لديهم خبرة عملية أكثر منك.

مسرد المصطلحات

الإطار - مجموعة معينة من أدوات التطوير التي يمكن الوصول إليها عبر واجهة برمجة التطبيقات الخاصة بها.

التأطير السلكي - خطوة مهمة في عملية تصميم الشاشة، حيث تساعد المطور على تخطيط الشكل النهائي وتصفح المستخدم باستخدام النماذج الورقية أو الإلكترونية للأجهزة ومكوناتها المرئية.

الدالة أو الإجراء - هي مجموعة من التعليمات البرمجية، يفضل أن تكون بين 5 و 50 سطرًا، لها غرض محدد واحد. على الرغم من كتابتها مرة واحدة فقط، يمكن تنفيذها عدة مرات خلال برنامج باستخدام "استدعاء الدالة"، مما يقلل من الحاجة إلى تكرار التعليمات البرمجية. في بعض لغات البرمجة، يتم استخدام مصطلحي "الدالة" و"الإجراء" بالتبادل، ولكن في البعض الآخر، هما مفهومان مختلفان للغاية: عادةً ما تعيد الدالة قيمة محسوبة، بينما يقوم الإجراء بمهمة واحدة قابلة لتعرفها.

جدول التتبع - جدول يتم إنشاؤه بواسطة المطور من خلال اختبار الصندوق الأبيض، والذي يرسم التغييرات في متغيرات البرنامج عند تنفيذ حالة الاستخدام. سيكون لجدول التتبع إدخالات لكل حالة استخدام مع النتائج المتوقعة (ما يجب أن يحدث) والنتائج الفعلية (ما يحدث بالفعل). يساعد مقارنة هذه النتائج المطور على تقييم ما إذا كان التطبيق يعمل بشكل صحيح.

جهاز Android الافتراضي (AVD) - إصدار تمت محاكاته ببرنامج لجهاز مادي معين، يعمل داخل نظام تشغيل مضيف مثل Microsoft Windows أو Apple OS X.

خطأ وقت التشغيل - مشكلة تحدث في أثناء استخدام التطبيق، ما يؤدي عادةً إلى قفله (رفض قبول إدخال المستخدم) أو تعطله (إنهاء المستخدم وإعادته إلى قائمة الجهاز أو الشاشة الرئيسة).

شاشات اللمس السعوية - تستخدم طبقتين متباعدتين من الزجاج المطلي بمكثفات أكسيد القصدير والإنديوم (ITO) الدقيقة. عندما يلمس إصبع المستخدم الشاشة، فإنه يغير المجال الكهروستاتيكي المحلي للشاشة. هذا النوع من شاشات اللمس أكثر سطوعًا وحساسية من شاشات اللمس المقاومة ويدعم إيماءات اللمس المتعدد. ومع ذلك، فإن التعقيد الهيكلي للشاشة يرفع تكاليف الإنتاج.

شاشات اللمس المقاومة - تستخدم طبقتين (عادة الزجاج والبلاستيك) مغطى بمواد موصلة للكهرباء (عادة أكسيد الإنديوم القصديري (ITO)). يتم إبقاء الطبقتين منفصلتين حتى يضغط إصبع أو قلم على بعضهما البعض، مما يتسبب في تغيير موضعي في المقاومة الكهربائية. هذا النوع من الشاشات اللمسية أقل تكلفة في التصنيع من الشاشات اللمسية السعوية ولكنه ليس حساسًا للغاية ولا يمكنه دعم إيماءات اللمس المتعدد.

العامل - رمز خاص (أو رموز متعددة) يخبر البرنامج بإجراء عمليات حسابية أو ارتباطية أو منطقية محددة على بياناته. يجب استخدام العوامل بترتيب أولوية محدد (يصف ذلك ما يتم تنفيذه أولًا). قد تكون معتادًا على هذا المفهوم من استخدام BIDMAS (أو BODMAS) (الأقواس، الأسس/الترتيب القسمة، الضرب، الجمع والطرح) في الرياضيات.

كائنية التوجه - جافا، التي نستخدمها لإنشاء تطبيقات أندرويد، مصنفة ضمن لغات البرمجة الكائنية التوجه (OOP) القائمة على الفئات. في لغات البرمجة الموجهة للكائنات، يتم إنشاء الكائنات من فئات يتم تصميمها عادةً على غرار (الأشياء) الواقعية. تعمل كل فئة كخريطة برمجية، حيث تقوم بتغليف (أو احتواء) حالة الشيء (بياناته أو خصائصه) وسلوكياته (وظائفه أو طرقه) في كود البرنامج.

لغات البرمجة كائنية التوجه – هي نهج حديث في البرمجة (نموذج) لتطوير البرمجيات، يعمل عن طريق نمذجة المشكلات الواقعية وتبسيط العمليات المعقدة إلى تفاعلات أساسية بين الكائنات المختلفة، مثل العلاقة بين العميل وحسابه البنكي.

مجموعة تطوير البرامج (SDK) - مجموعة من أدوات البرامج المقدمة للمطورين لإنشاء تطبيق لمنصة معينة، على سبيل المثال Java SDK لنظام Android. عادةً ما يحتوي SDK على واجهات برمجة التطبيقات (APIs)، وأدوات مثل المترجمات وأدوات تصحيح الأخطاء، ووثائق مرجعية، ونماذج من الكود.

محسّنة - يتم إنشاء الأصول المحسّنة باستخدام تنسيقات ملفات أكثر كفاءة لأنها تتطلب مساحة تخزين أقل. يمكن أن يؤدي ذلك إلى تحسينات في الأداء ومساحة رقمية أصغر على موارد الجهاز. من الأمثلة على ذلك استخدام صور .jpg بدلاً من ملفات .bmp حيث تستخدم هذه الأخيرة ضغط البيانات لتقليل حجم الملف.

معالجة الأحداث – تصف عملية استخدام دالة مكتوبة خصيصًا أو طريقة من طرق الفئة لتنفيذ إجراءات محددة عند تفعيل حدث من قبل المستخدم أو النظام. على سبيل المثال، يمكن عرض نتيجة عملية حسابية عندما يتم الضغط على زر، أو عرض مؤشر "البطارية منخفضة" على الجهاز عندما ينخفض مستوى الشحن إلى حد معين.

نوع البيانات - نوع البيانات التي نريد تخزينها في ذاكرة الوصول العشوائي للجهاز المحمول. يتطلب تخزين كل نوع من البيانات كميات مختلفة من ذاكرة الوصول العشوائي. على سبيل المثال، يختلف نوع البيانات المطلوب لتخزين رقم عن النوع المستخدم لتخزين حرف وسيتطلب أحجامًا مختلفة من ذاكرة الوصول العشوائي. تختلف الأسماء الدقيقة لأنواع البيانات بين لغات البرمجة المختلفة، لذا احرص على اختيار اللغة الصحيحة.

واجهة برمجة التطبيقات (API) - تعمل كمكتبة من الإجراءات الروتينية المكتوبة مسبقًا والتي توفر للمطور إمكانية الوصول إلى أنظمة أخرى (مثل قواعد البيانات) أو نظام تشغيل الجهاز أو أجهزته الفعلية. الهدف من واجهة برمجة التطبيقات هو تسهيل عملية التطوير من خلال التخلص من (إخفاء) تعقيد أنظمة البرامج والأجهزة الموجودة تحتها.