

Выбор второй особи для скрещивания возможен по следующим вариантам:

1) случайный выбор. Вторая особь выбирается случайно из родительской популяции. Данный вариант наиболее прост для реализации, однако вследствие своей природы не может гарантировать качество получаемого решения;

2) родственное скрещивание на основе анализа расстояния между особями. В качестве расстояния между хромосомами для решаемой задачи целесообразно использовать расстояние Хэмминга, определяемое для двух особей [3].

В соответствии с этим различают близкородственное (инбридинг) и дальнеродственное (аутбридинг) скрещивание.

При аутбридинге выбирается хромосома с максимальным значением расстояния Хэмминга до первой выбранной. Данный вид отбора обеспечивает снижение скорости сходимости алгоритма к локальному экстремуму и увеличивает вероятность достижения глобального экстремума.

При инбридинге выбирается хромосома с минимальным расстоянием от первой выбранной. Данный вид отбора обеспечивает максимально быструю сходимость алгоритма к локальному экстремуму.

Для обеспечения сохранения лучших найденных особей содержимое элитной популяции автоматически включается в дочернюю на текущем шаге алгоритма.

Размер элитной популяции L^{el} определяется как

$$L^{el} = k^{el} \cdot L^{sr},$$

где k^{el} — коэффициент элитизма; L^{sr} — размер исходной популяции.

Операция скрещивания выполняется над парой хромосом, являющихся родительскими. Результатом являются две хромосомы-потомка.

Экспериментальная проверка показала, что одним из наиболее эффективных с точки зрения конечного результата является универсальный кроссинговер [4].

При решении задачи о ранце рассматривались две схемы генерации особей для формирования исходных популяций: случайная генерация варианта решения (особи) с последующей проверкой выполнимости ограничений и последовательная генерация варианта решения (особи) с поэтапной проверкой выполнимости ограничений.

Заключение. Обе схемы основаны на случайной генерации значений переменных в варианте решения задачи о ранце.

Первая схема потенциально имеет больше шансов сформировать уже в начальной популяции оптимальное решение, однако требует существенного отсева вариантов, не удовлетворяющих ограничениям.

В соответствии со второй схемой, если на последующем этапе полученные значения переменных в варианте решения задачи о ранце не удовлетворяют ограничениям, то за вариант решения берётся особь, полученная на предыдущем этапе. Данная схема формирует исходную популяцию за количество шагов, незначительно превышающее размер популяции.

Список цитируемых источников

1. Holland J. H. *Adaptation in Natural and Artificial Systems*. Cambridge : MIT Press, 1992. 228 p.
2. Гладков Л. А., Курейчик В. В., Курейчик В. М. *Генетические алгоритмы*. М. : ФИЗМАТЛИТ, 2006. 320 с.
3. Рутковская Д., Пилиньский М., Рутковский Л. *Нейронные сети, генетические алгоритмы и нечёткие системы*. М. : Горячая линия — Телеком, 2004. 383 с.
4. Нгуен Минь Ханг. Применение генетического алгоритма для задачи нахождения покрытия множества // Тр. ин-та систем. анализа РАН, 2008. № 33. С. 206—219 ; Панченко Т. В. *Генетические алгоритмы : учеб.-метод. пособие*. Астрахань : Астрахан. ун-т, 2007. 87 с.

УДК 004.428

А. А. Зданович, А. В. Шах

Учреждение образования «Барановичский государственный университет», Барановичи

КРОССПЛАТФОРМЕННАЯ РАЗРАБОТКА МОБИЛЬНЫХ ПРИЛОЖЕНИЙ С ПРИМЕНЕНИЕМ XAMARIN

В данной работе представлены результаты исследования фреймворка Xamarin.

This paper presents the results of research framework Xamarin.

Введение. Разработка мобильных приложений заняла важное место в сфере информационных технологий. Каждая компания стремится создать мобильную версию своего сервиса. В некоторых

сегментах рынка мобильное продвижение — это, фактически, обязательное условие для существования и развития бизнеса. Мобильное приложение — современный высокоэффективный бизнес-инструмент, с помощью которого коммерческое предложение оперативно доносится пользователю. Стоимость разработки мобильных приложений зависит от сложности и сроков изготовления.

Основная сложность в разработке мобильных приложений заключается в многообразии операционных систем. Для каждой мобильной платформы требуется знание разных технологий и языков программирования, за счёт чего увеличиваются стоимость и время разработки. Однако в феврале 2013 г. американская компания Xamarin выпустила продукт, позволяющий разрабатывать кроссплатформенные мобильные приложения. С использованием данного продукта для разработки мобильных приложений специалистам достаточно знания одного языка программирования, что существенно сокращает стоимость и время разработки.

Основная часть. Мобильные технологии присутствуют в большинстве сфер нашей жизни. На данный момент можно выделить три основные мобильные операционные системы: Android, iOS, Windows Phone. Это вызывает потребность в разработке приложений одинакового функционала на все три платформы.

Xamarin — фреймворк, предназначенный для разработки кроссплатформенных мобильных приложений. Разрабатывая приложение на C# и .NET, воспользовавшись Xamarin, разработчик получает возможность легко портировать приложение практически на любую мобильную платформу (iOS, Android, Windows Phone 8). Код пишется на языке C# с применением привычных средств: LINQ, лямбда-выражений, Generic и async. При этом имеется полный доступ ко всем возможностям SDK платформы и механизму создания UI, получая на выходе приложение, которое ничем не отличается от нативных и не уступает им в производительности.

Инсталлятор Xamarin устанавливает плагин к Visual Studio, который позволяет разрабатывать приложения для популярных мобильных платформ в привычном разработчику окружении. Также устанавливается отдельная среда разработки Xamarin Studio, которая является модифицированной версией MonoDevelop.

После установки в Visual Studio добавляются шаблоны проектов для мобильных приложений под Android и iOS (поддерживается создание как специализированных приложений для iPad и iPhone, так и универсальных приложений, а также приложений, использующих OpenGL).

Фреймворк состоит из нескольких основных частей: 1) Xamarin.iOS — библиотека классов для C#, предоставляющая разработчику доступ к iOS SDK; 2) Xamarin.Android — библиотека классов для C#, предоставляющая разработчику доступ к Android SDK; 3) Xamarin.Forms — библиотека классов для C#, предоставляющая разработчику доступ к iOS SDK, Android SDK, Windows Phone SDK; 4) Компиляторы для iOS и Android; 5) IDE Xamarin Studio; 6) Плагин для Visual Studio [1].

В Xamarin.Forms создаётся пользовательский интерфейс, описывается в общей для всех платформ части. Он является надстройкой над уже существовавшими до него Xamarin.iOS, Xamarin.Android и Xamarin.WinPhone и представляет собой набор редакторов, навигационных панелей, лейаут панелей и т. д. (рисунок 1).

Преимущества Xamarin.Forms: использование всех возможностей C# и .NET; подход к созданию и работе с пользовательским интерфейсом близок к тому, что применяется в Windows, поддерживает работу с XAML, биндинги, темплейты и т. д.; имеется возможность использования кода, написанного ранее под систему Winwos; визуальные элементы имеют BindingContext (аналог DataContext в WPF), есть BindableProperty (аналог DependencyProperty), можно связывать View с ViewModel аналогично тому, как в WPF; если UI описывается только в одном месте, то приложения под разными системами будут выглядеть очень похоже, что может быть важно, например, в корпоративных разработках.

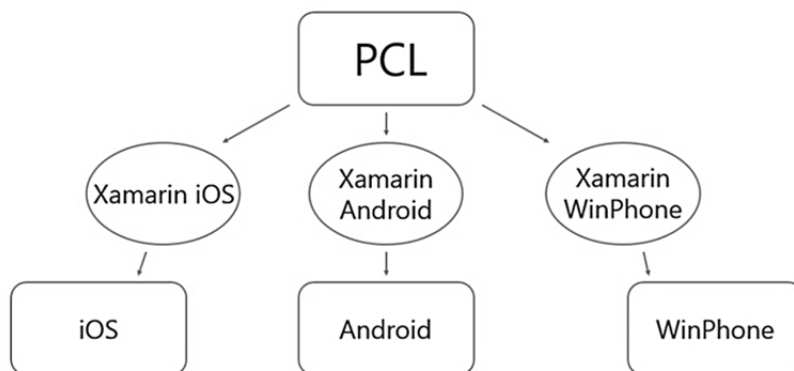


Рисунок 1 — Схема работы Xamarin.Forms [2]

Можно выделить следующие группы проблем, с которыми приходится сталкиваться при разработке на Xamarin.Forms: неполная реализация функционала WPF; компромиссные решения в реализации функционала, различающегося на разных платформах; различное поведение на разных платформах.

На момент написания статьи Xamarin имеет следующие типы лицензий:

– Starter (бесплатный. Предназначен скорее для ознакомления, так как имеет ограничение на размер приложения и использование сторонних компонентов);

– Indie (300 дол. США. Рассчитан для частного пользования. Снимается ограничение на размер приложения. Разработка возможна только в Xamarin Studio);

– Business (999 дол. США. Предназначен для организаций. Появляется возможность разработки в Visual Studio и приватная техническая поддержка от инженеров Xamarin);

– Enterprise (1899 дол. США. Рассчитан для организаций. В рамках этой лицензии предоставляется возможность получения Hotfixes, а также отправки инженерам проекта с исходным кодом) [3].

Для тестирования функций Xamarin было написано приложение под системы Android, iOS, Windows Phone, демонстрирующее работу с базой данных (рисунок 2).

Заключение. Технология Xamarin является серьёзным инструментом для решения сложных задач в области разработки мобильных приложений. За последние два года заметны серьёзные улучшения в общей стабильности продукта. Однако довольно высокая стоимость лицензии приложения может стать препятствием для его использования.

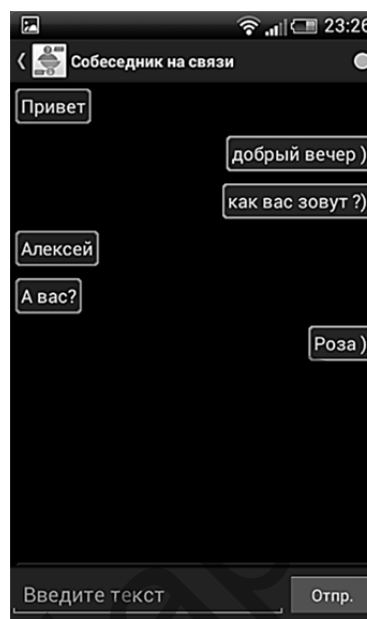


Рисунок 2 — Вид приложения на Android

Список цитируемых источников

1. Xamarin is a better way to. URL: <https://xamarin.com> (дата обращения: 21.09.2015).
2. Там же.
3. Xamarin. За и против. URL: <http://habrahabr.ru/post/227863> (дата доступа: 21.09.2015).

УДК 004.93*12

А. И. Калько

Учреждение образования «Барановичский государственный университет», Барановичи

О. И. Наранович,

кандидат физико-математических наук, доцент

Учреждение образования «Барановичский государственный университет», Барановичи

ОБНАРУЖЕНИЕ И СЛЕЖЕНИЕ ЗА ОБЪЕКТАМИ ПО ИХ ЦВЕТУ С ПРИМЕНЕНИЕМ БИБЛИОТЕКИ OPENCV

Текст публикации не представлен по решению авторов.