



## Обучение студентов разработке ГИС-приложения

**Наталья Дмитриевна Хрулёва**

ООО «Сатурн-центр»  
ведущий программист 1С  
nataliia@khruleva.ru  
 0000-0003-0000-0000

Поступила в редакцию 29.07.2021  
Принята 19.09.2021  
Опубликована 15.10.2021

 10.25726/w0687-1762-2053-o

### Аннотация

Мобильные ГИС-приложения становятся все более сложными, как решаемые с их помощью задачи. Обычное ГИС-приложение должно включать такие элементы, как искусственный интеллект, распознавание образов или машинное обучение, реляционные или нереляционные базы данных, пространственное представление и рассуждения. Такие компании, как Google и Apple, разрабатывают новые технологии, связанные с разработкой мобильных приложений. Например, Apple представила в 2019 году на WWDC2019 и WWDC2020 новую технологию под названием SwiftUI, которая направлена на снижение сложности разработки мобильного приложения и позволяющая интегрировать такие технологии, как MapKit, для представления пространственной информации. В данной работе представлены исследования преимуществ использования SwiftUI для интеграции MapKit в качестве основы пространственного представления для облегчения разработки мобильных ГИС-приложений. Информационные технологии имеют большое разнообразие применений в различных областях науки. Например, искусственный интеллект и машинное обучение - это технологии, которые начинают широко использоваться в мобильных приложениях. Целью данной работы является исследования способов разработки мобильных приложений, которые могут выполнять представление и вычисления информации в соответствии с требованиями.

### Ключевые слова

пользовательский интерфейс, SwiftUI, MapKit, ГИС, пространственное представление.

### Введение

Согласно работы (Inie, 2020), вычислительная система состоит из трех уровней или слоев: Вычислительной теории, Представления информации или знаний, в виде разработанного и реализованного алгоритма, и наконец, Аппаратной реализации. Также можно добавить реализацию программного обеспечения на выбранном языке программирования и используемую технологическую платформу. В последнем нам необходимо рассмотреть технические элементы для улучшения пространственной и временной производительности алгоритма, уменьшения ошибок, планирования транзакций блоков данных, доступа к общей или распределенной памяти и т.д. (Dix, 2017). Таким образом, мы можем рассматривать эти слои как архитектуру, которую можно использовать в качестве основы (рис.1).

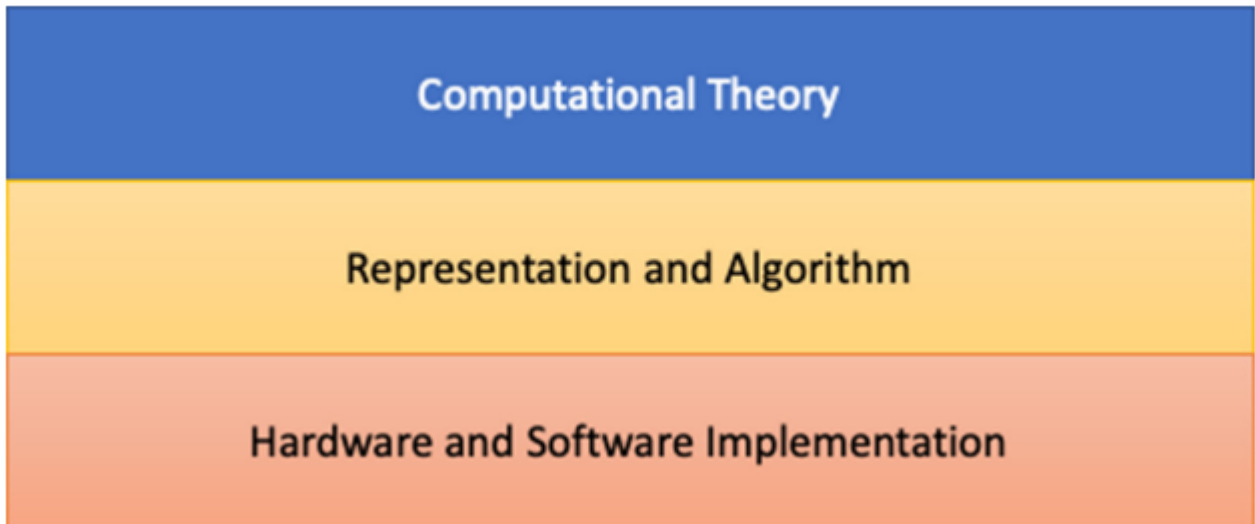


Рисунок 1. Архитектура вычислительной системы

Одной из целей многоуровневой архитектуры является возможность разделения действий, для снижения сложности модулей и их повторного использования. Связь между слоями должна быть простой и легкой. В качестве примера можно использовать архитектуру Open System Interconnection, где каждый уровень предоставляет одну услугу. Например, в 5-слойной гибридной модели. Слои предоставляют услугу упаковки, маршрутизации по сети, синхронизации и кодирования кадров (Mcgill, 2020). Наконец, связь между слоями осуществляется с помощью интерфейса. На рис. 2 представлена общая модель слоев многоуровневой архитектуры.

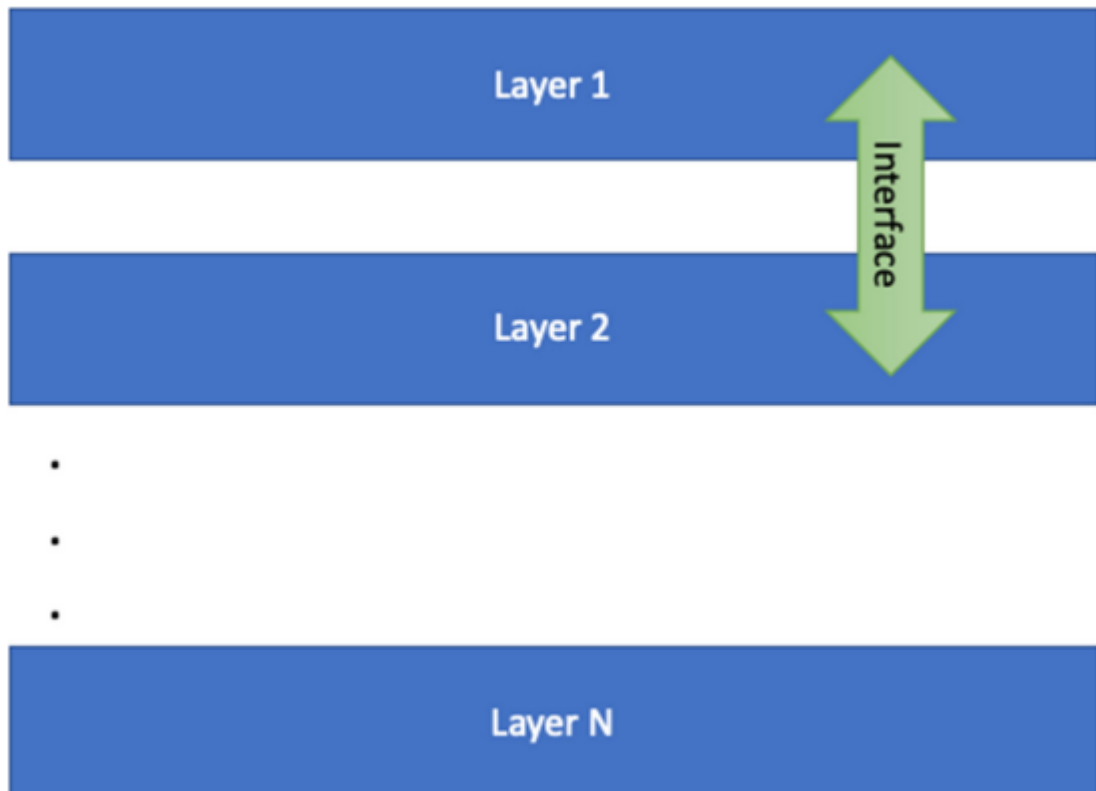


Рисунок 2. Слои многослойной архитектуры согласно (Heer, 2010)

Как упоминалось ранее, каждый слой выполняет определенную функцию. Эти функции должны быть прозрачными для других слоев. Таким образом, интерфейсы являются ключом к общению слоев. Платформы разработки, такие как Android, Eclipse и XCode имеют графический пользовательский интерфейс (GUI), который облегчает разработку программного кода, позволяет сосредоточиться на программировании и повысить производительность (Пономарев, 2020). Данные платформы делают процесс разработки более интуитивным и понятным. В 2019 году была представлена новая технология под названием SwiftUI, которая была интегрирована в платформу Xcode.

Декларативная модель в настоящее время используется в основных технологиях, таких как Xamarin, UWP и WPF с XAML с целью упрощения задач. В 2020 году Apple анонсировала новые интерфейсы с новыми функциями для режима стека на WWDC 2020.

Эти интерфейсы обеспечивают возможность постепенного добавления функциональных возможностей, позволяя разрабатывать алгоритмы в виде блоков Lego. Одной из интересных особенностей этих интерфейсов является возможность избежать и уменьшить необходимость управления техническими деталями.

### **Материалы и методы исследования**

Предлагаемая, в данной работе, методология относится к гибким (agile) методологиям. Предлагается сосредоточиться на разработке трех частей, которые являются дизайном, реализацией и обратной связью. Преимущества такого подхода заключаются в том, что можно постепенно интегрировать модули в архитектуру, например, перенастроить презентацию или добавить другую функциональность в приложения. Предлагаемая методология позволяет быстро реконфигурировать приложения, независимо от того, нужно ли добавлять элементы в функциональность ГИС или менять технологию с MapKit на ArcGIS, не затрагивая другие элементы проекта (рис. 3).

Разработка программного кода XCode ведется при помощи специального механизма, который называется storyboard. Реализация данного механизма обеспечивается при помощи файла под названием Main.storyboard, который представляет структуру объектов.

Файл ViewController.swift управляет поведением приложения. Xcode отображает графическое представление из файла Main.storyboard. Все графические элементы могут вставляться в этот файл. Проект может быть скомпилирован для запуска в симуляторе или на внешнем устройстве. Xcode позволяет выбрать конкретное устройство, на котором можно выполняться приложение. Например, iPad, iPad Pro, iPhone X, iPhone 8 и т.д.

Проблема с компиляцией каждый раз заключается в том, что если необходимо увидеть незначительные изменения, то необходимо потратить много времени на повторную компиляцию.

Предположим, что наше приложение должно добавить географическое представление. Для этого можно MapKit. Можно разделить приложение на четыре части. Одна часть может описывать распределение элементов в телефоне, аналогично HTML. Затем добавляем функциональность или некоторые другие элементы в приложение. Далее есть модуль, который управляет географическим представлением и получает географические данные, долготу и широту. Наконец, слой, который обрабатывает данные. На рис. 4 представлена архитектура, которую можно использовать для добавления нужной функциональности.

Для работы с Картой имеется файл, содержащий весь код, связанный с управлением Картами. В данном случае это файл MapView. Импорт библиотек SwiftUI и MapKit позволяет расширить класс UIViewRepresentable. В SwiftUI можно добавлять новые местоположения.

Дополнительная функциональность можно добавить и описать в swift-файле - SwiftUIView.swift. Новый добавляем функционал представляет собой добавление нового изображения. В результате получаем быстрый пользовательский интерфейс и его интеграцию с технологией MapKit. В данном примере обработка данных является статической. Но для других приложений можно использовать базу данных, например файл GeoJSON для хранения данных на устройстве и т.д. Структура данных, для хранения получаемой информации, использует модель класса - DataModel.swift. Файл должен импортировать библиотеку MapKit, для использования объектов CLLocationCoordinate и MKAnnotation.

Для SwiftUI в структуре МКMapView используется функция `updateUIView()`. Эта функция обрабатывает карту. Она дает возможность устанавливать и получать элементы из МКMapView (такие как долгота, широта, регион, масштабирование).

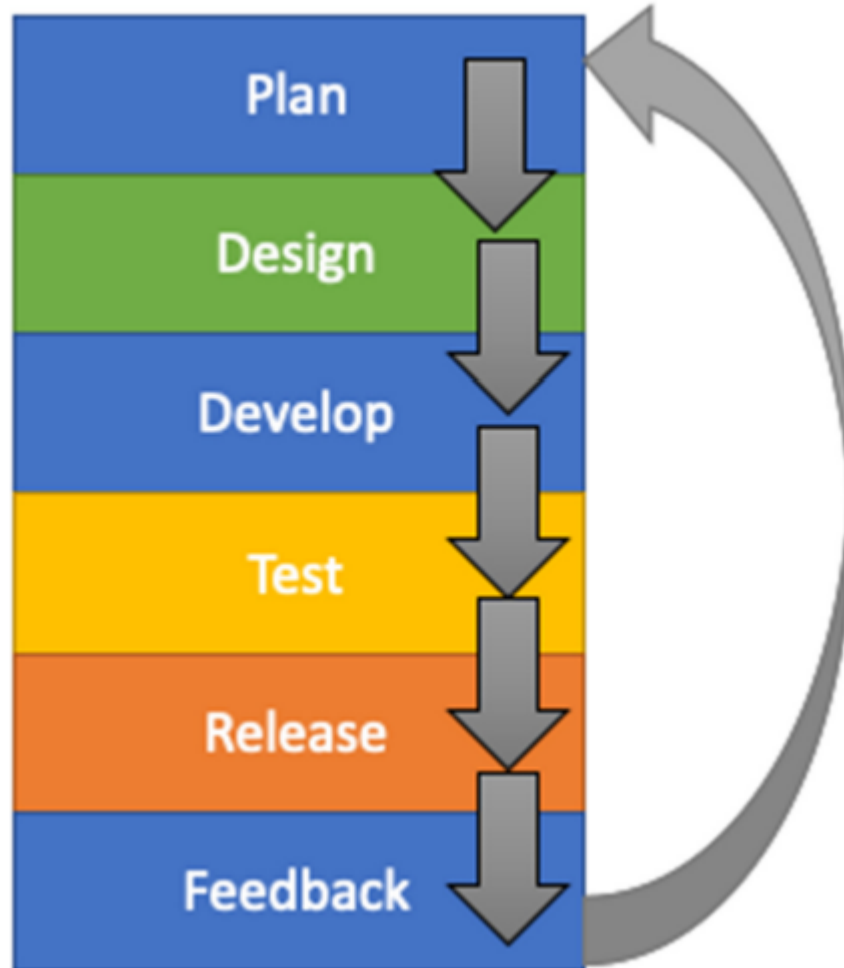


Рисунок 3. Предлагаемая гибкая методология разработки

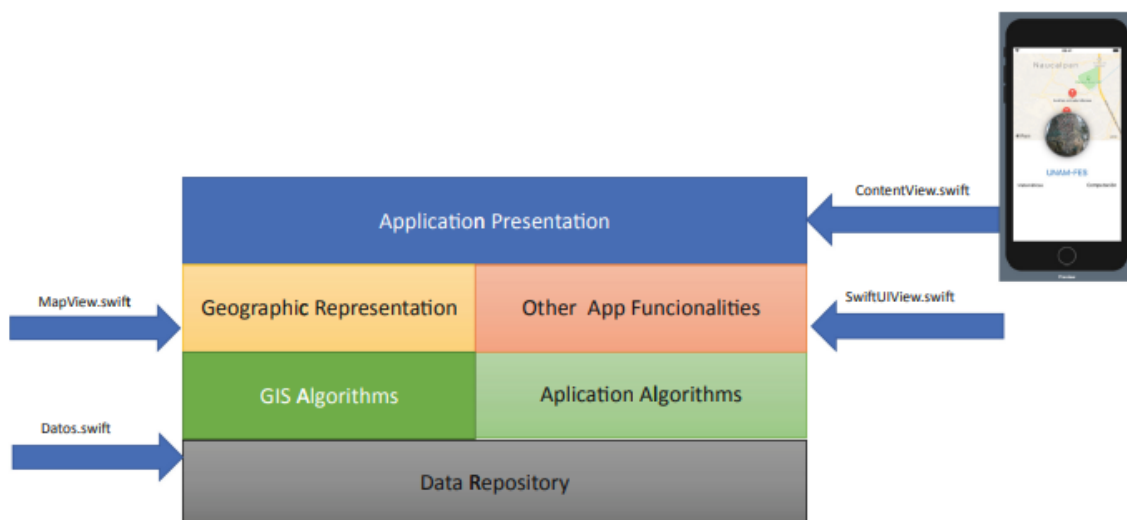


Рисунок 4. Предлагаемая архитектура мобильного ГИС-приложения

В случае с Main.storyboard используются те же методы и свойства (регион, местоположение, диапазон). Функция viewDidLoad() позволяет загружать карту после выполнения приложения. Разница в том, что нам нужно вручную добавить розетку для передачи Main.storyboard в ViewController.swift.

### Результаты и обсуждение

Обновление средств ИКТ, операционных систем, усовершенствование системного и прикладного программного обеспечения, появление новых языков программирования ставят перед системой образования новые задачи. При таких условиях важным является научно обоснованное, взвешенное и методически правильное использование веб-ориентированных сред обучения программированию в учебных заведениях в целом и в учреждениях общего среднего образования в частности (Шадрина, 2016).

Использование метода экспертного оценивания для выделения наиболее значимых веб-ориентированных сред обучения программированию в учреждениях общего среднего образования заключается в установлении соответствующем среде баллов по определенным критериям и сортировке по данному критерию. В целом, на рассмотрение экспертов было предложено 5 различных веб-ориентированных сред обучения программированию, которые могут быть использованы в учреждениях общего среднего образования. Соглашаясь с мнением ученых (Сербин, 2020), под критериями отбора веб-ориентированных сред обучения программирования будем понимать такие качества, признаки и свойства веб-ориентированных технологий, которые необходимы для успешного обучения основам программирования учащимися учреждений общего среднего образования.

Взяв за основу общий подход к критериям отбора средств обучения, была предложена балльная система ранжирования, по которой для  $N=5$  сред обучения программированию эксперты предоставляли соответствующих значений. Чтобы выяснить степень проявления каждого критерия эксперты оценивали его показатели. Оценивание происходило по следующим параметрам: от 0 – показатель отсутствует до определенного установленного максимального значения – показатель проявляется в полной мере. Данное максимальное значение показателя определялось как важность данного параметра в пределах применения к образовательному процессу. Как отмечают исследователи (Alpaydin, 2021), важным является то, что количество показателей для определения степени весомости критериев отбора веб-ориентированных сред обучения программирования может оказаться большим, а отдельные показатели определенного критерия могут быть недостаточно значимыми для его отбора.

Выбор представленных сред обучения был обусловлен анализом практического опыта учителей информатики во время изучения курса программирования. Однако, в процессе анализа выяснилось, что существует определенное количество веб-ориентированных сред обучения программированию, доля использования которых является относительно небольшой. Приведем некоторые из них stepic.org, edx.org, hourofcode.com, scratch.mit.edu. И хотя данные среды остались вне данному исследованию, они требуют дальнейшего анализа на предмет их эффективного использования.

Согласно методике исследования, чтобы исключить психологическое давление на экспертов, который бы мог повлиять на выбор через установленный порядок ранжирования, веб-ориентированные среды обучения программированию на форме размещались отсортированы по возрастанию в алфавитном порядке, что отражено в таблице 1.

Таблица 1. Веб-ориентированные среды обучения

Название	Описание	Стоимость
code.org	Изучение основ программирования для детей в игровой форме	0
codeschool.com	Профессиональный инструмент для изучения широкого спектра ИТ технологий	10 дней (200 мин) бесплатно От 29\$ / месяц
e-olymp.com	Проверка навыков олимпиадного программирования; организация соревнований и турниров	0

javarush.ru	Изучение программирования на языке Java на высоком теоретическом и практическом уровне	30\$/месяц 50\$/месяц
sololearn.com	Мобильное приложение для изучения основ программирования на разных языках	Обычная 0 PRO 4,99\$ / месяц

Таким образом, осуществив сравнение пяти веб-ориентированных сред обучения программированию по установленным критериям и показателям отбора, можно констатировать важную роль объектов исследования в учебном процессе. По нашему мнению, их использование является эффективным средством формирования и развития информационно-коммуникационной компетентности. Стоит обратить внимание на высокую оценку экспертами эргономичности представленных сред, что минимизирует сроки их освоения; многоязычия интерфейсов, а также экономическую доступность. В частности стоит рекомендовать sololearn.com и code.org для изучения основ программирования, а e-olymp.com для углубления знаний в решение задач на разных языках программирования и организации олимпиад, соревнований и конкурсов.

Перспективным видим разработку методических рекомендаций по использованию веб-ориентированных сред обучения программированию, а также в популяризации их среди педагогического сообщества путем проведения семинаров, вебинаров, мастер-классов и включив в программу курсов повышения профессионального мастерства. В дальнейших исследованиях следует обоснованно расширить критерии отбора и их показатели, а также проанализировать другие имеющиеся веб-ориентированные среды обучения программированию.

Проведем результаты анкетированного исследования.

Анкетирование, проведенное среди студентов-первокурсников непосредственно перед изучением курса «Программирование» показало, что около двух третей опрошенных изучали в школе язык программирования Python, как основной. Некоторые студенты изучали несколько языков программирования (преимущественно самостоятельно, а не в рамках школьного курса). Вместе с тем, с другой задачей анкеты – написанием на любом известном языке фрагмента кода для обмена значениями двух переменных-справились лишь 3 из 17 студентов. То есть более 80% первокурсников не владеют ни одним языком программирования даже на начальном уровне. При таких обстоятельствах язык программирования, который будет использоваться в рамках данного курса, по сути, станет первым для большей части группы. При использовании же языка, который не изучался ни одним из опрошенных, студенты с более высоким уровнем подготовки не будут терять мотивацию к изучению курса разработки мобильного приложения благодаря элементу новизны.

Тем более, что первоочередной задачей курса "Программирование" для будущих специалистов по направлению подготовки "Информатика" является закладка основ профессиональной деятельности (развитие алгоритмического мышления, формирование стиля программирования, способности находить эффективные решения поставленных задач и тому подобное).

Язык программирования Python предоставляет возможность "быстрого старта". Написание первых программ требует минимальных знаний синтаксиса языка программирования. Эти знания будут углубляться по мере усложнения учебных задач. Простая задача, как правило, будет иметь простое решение на Python. Самые первые задания для большей наглядности целесообразно выполнять в режиме командного интерпретатора.

Еще одним несомненным преимуществом Python является то, что это современный инструмент, который имеет широкую сферу применения и большой набор прикладных библиотек. Python доступен на многих платформах, существуют также online-интерпретаторы, не требующие предварительной установки. То есть использование Python для обучения студентов направлению подготовки "Информатика" решает не только учебные, но и профессиональные задачи, которые в будущем предстанут перед выпускником.

Наконец, форматирование кода как требование синтаксиса Python сразу приучает студентов структурировать код, что является важной задачей именно на первых этапах обучения программированию.

Кроме того знание основ языка программирования Python может понадобиться при будущем изучении курса "методы вычислений" с использованием системы компьютерной алгебры Sage, при изучении курса веб-программирования, моделирования и других.

Использование Python как первого языка программирования имеет и свои недостатки, среди которых существенным является отсутствие статической типизации. Поэтому следует уделить особое внимание введению понятия типа данных.

Логика построения лабораторных работ является вполне типичной для курса программирования, хотя некоторые задачи построены таким образом, что побуждают использовать специфические возможности Python (как то умножение строке на число). Часть задач была разработана на основе популярных сборников задач (Ильин, 2017), другие же являются авторскими разработками. Каждая лабораторная работа состоит из общей части (задачи, должны быть выполнены всеми студентами) и вариативной части (задания по вариантам). Каждая лабораторная работа содержит 20 вариантов заданий.

Помимо лабораторных работ курс предусматривает выполнение двух контрольных работ. Первая – аудиторная – проводится в середине семестра и содержит несколько вариантов заданий для проверки уровня усвоения материала. Вторая контрольная работа представляет собой небольшой учебный проект, который студенты выполняют в течение семестра. Проекты предусматривают более детальное ознакомление с какой-нибудь из библиотек Python на выбор студента (графические расширения, математические расширения и тому подобное).

Составляющими оценки за лабораторную работу являются: выполнение работы, защита работы, отчет о выполнении, качество выполнения, соблюдение графика. Поскольку данный курс фактически является вступительным к специальности, то большое внимание уделяется стилю программирования. Составляющая "отчет о выполнении" предусматривает штрафы (в процентах от баллов, которые начисляются за лабораторную работу) за плохое комментирование кода или отсутствие коротких объяснений в отчете о выполнении лабораторной работы. Составляющая "выполнение работы" допускает снижение оценки за неинформативный вывод результатов работы программы, отсутствие и технологического образования, фильтрация входных данных. Составляющая "качество исполнения" предусматривает бонусы за реализацию удобного интерфейса пользователя, использование программных конструкций, которые оптимизируют код по критериям быстродействия или объема используемой памяти.

Для изучения теоретического материала как необходимый минимум было рекомендовано пособие (Парыгин, 2019) из-за его компактности и доступности (материал адаптирован для общеобразовательной школы). Недостатком данного пособия является то, что все примеры в нем написано на Python 2.x. Для более детального изучения рекомендовались классические учебники (по авторству Гвидо Ван Россума и Марка Лутца).

По результатам внедрения курса можно сказать, что он в целом выполняет поставленные перед ним задачи. Курс не только формирует базовые навыки программирования, но и вооружает будущих специалистов современным мощным инструментом, владение которым повышает спрос на специалиста на рынке труда.

В будущем планируется дополнить практикум работами, ориентированными на использование коллекций объектов Python, включить в работы краткий теоретический материал с примерами на Python 3.x и повысить интеграцию с другими курсами.

### **Заключение**

Использование Python/SwiftUI дает возможность создавать простые и быстрые мобильные приложения. SwiftUI предлагает высокий уровень параметров разработки, таких как WYSIWYG. Данная характеристика позволяет отображать результаты реализации программного кода для пользовательского практически в режиме реального времени. Это позволяет быстро просматривать и тестировать функциональность приложения. С другой стороны, очевидно, что более функциональное приложение потребует большей степени выразительности, которую может предложить Main.storyboard.

Цель использования этой структуры состоит в том, чтобы использовать SwiftUI в качестве исследовательской лаборатории в которую интегрируются функциональные возможности с основным приложением или системой, изолируя возможные ошибки. Таким образом можно сделать вывод, что SwiftUI предлагает возможности по разработке функционального пользовательского интерфейса.

Предлагаемая в работе разделенная на уровни архитектура, может оказаться полезной для отслеживания ошибок и добавления новых функций, которые при этом не затрагивают другие модули.

### Список литературы

1. Зарипова Г., Рузиева К. Использование интерактивных методов в процессе обучения студентов компьютерным технологиям // Проблемы педагогики. 2018. № 7 (39). С. 5-8.
2. Ильин М.С., Тарасов В.С. Анализ средств по ускорению вычислений для устройств на платформе android // Сборник студенческих научных работ факультета компьютерных наук ВГУ. 2017. С. 123-130.
3. Исмоилова М.Н., Имомова Ш.М. Интерполяция функции // Вестник науки и образования. 2020. № 3(81). Часть 3. С. 5.
4. Парыгин Д.С. Применение рекомендательных технологий в системах с пространственной информацией // Прикаспийский журнал: управление и высокие технологии. 2019. №1. С. 96-109.
5. Пономарев А.В. Решение задач линейного программирования с использованием GNU Octave, GLPK и Python. [https://cais.iias.spb.su/ponomarev/LP\\_tutorial.pdf](https://cais.iias.spb.su/ponomarev/LP_tutorial.pdf)
6. Сербин Д.В., Петров А.А. Машинное обучение в мобильной разработке // Цифровизация экономики: направления, методы, инструменты. 2020. С. 64-66.
7. Шадрин Н.И., Берман Н.Д., Решение задач оптимизации в Microsoft Excel 2010: учеб. пособие. Хабаровск: Изд-во Тихоокеан. гос. ун-та, 2016. 101с.
8. Alpaydin E. Machine learning. MIT Press, 2021.
9. Apple WWDC 2020. <https://developer.apple.com/videos/play/wwdc2020/10031>.
10. Dix, A.: Human-computer interaction, foundations and new paradigms. J. Vis. Lang. Comput. 42, 122–134 (2017)
11. Heer, J., Bostock, M.: Declarative language design for interactive visualization. IEEE Trans. Visual Comput. Graphics 16(6), 1149–1156 (2010)
12. Inie, N., Dalsgaard, P.: How interaction designers use tools to manage ideas. ACM Trans. Comput.-Hum. Interact. (TOCHI) 27(2), 1–26 (2020)
13. McGill, M., Kehoe, A., Freeman, E., Brewster, S.: Expanding the bounds of seated virtual workspaces. ACM Trans. Comput.-Hum. Interact. (TOCHI) 27(3), 1–40 (2020)
14. Microsoft 2020. Xamarin.Forms .NET. <https://dotnet.microsoft.com/apps/xamarin/xamarinforms>.
15. Tanenbaum, A., Wetheall, D.: Computer Networks. Pearson (2013)


### Teaching students to develop GIS applications

**Natalia D. Khruleva**

Saturn-Center LLC

leading 1C programmer


[khruleva@mail.ru](mailto:khruleva@mail.ru)

 0000-0003-0000-0000

Received 29.07.2021

Accepted 19.09.2021

Published 15.10.2021

 10.25726/w0687-1762-2053-o



### Abstract

Mobile GIS applications are becoming more and more complex, as the tasks they solve are. A typical GIS application should include elements such as artificial intelligence, pattern recognition or machine learning, relational or non-relational databases, spatial representation and reasoning. Companies such as Google and Apple are developing new technologies related to the development of mobile applications. For example, Apple introduced a new technology called SwiftUI at WWDC2019 and WWDC2020 in 2019, which aims to reduce the complexity of mobile application development and allows integrating technologies such as MapKit to represent spatial information. This paper presents studies of the advantages of using SwiftUI to integrate MapKit as a basis for spatial representation to facilitate the development of mobile GIS applications. Information technologies have a wide variety of applications in various fields of science. For example, artificial intelligence and machine learning are technologies that are beginning to be widely used in mobile applications. The purpose of this work is to investigate ways to develop mobile applications that can perform the presentation and calculation of information in accordance with the requirements.

### Keywords

user interface, SwiftUI, MapKit, GIS, spatial representation.

### References

1. Zaripova G., Ruzieva K. Ispol'zovanie interaktivnyh metodov v processe obuchenija studentov komp'yuternym tehnologijam // Problemy pedagogiki. 2018. № 7 (39). S. 5-8.
2. Il'in M.S., Tarasov V.S. Analiz sredstv po uskoreniju vychislenij dlja ustrojstv na platforme android // Sbornik studencheskih nauchnyh rabot fakul'teta komp'yuternykh nauk VGU. 2017. S. 123-130.
3. Ismoilova M.N., Imomova Sh.M. Interpoljacija funkcii // Vestnik nauki i obrazovanija. 2020. № 3(81). Chast' 3. S. 5.
4. Parygin D.S. Primenenie rekomendatel'nyh tehnologij v sistemah s prostranstvennoj informaciej // Prikaspijskij zhurnal: upravlenie i vysokie tehnologii. 2019. №1. S. 96-109.
5. Ponomarev A.V. Reshenie zadach linejnogo programmirovaniya s ispol'zovaniem GNU Octave, GLPK i Python. [https://cais.iias.spb.su/ponomarev/LP\\_tutorial.pdf](https://cais.iias.spb.su/ponomarev/LP_tutorial.pdf)
6. Serbin D.V., Petrov A.A. Mashinnoe obuchenie v mobil'noj razrabotke // Cifrovizacija jekonomiki: napravlenija, metody, instrumenty. 2020. S. 64-66.
7. Shadrina N.I., Berman N.D., Reshenie zadach optimizacii v Microsoft Excel 2010: ucheb. posobie. Habarovsk: Izd-vo Tihookean. gos. un-ta, 2016. 101s.
8. Alpaydin E. Machine learning. MIT Press, 2021.
9. Apple WWDC 2020. <https://developer.apple.com/videos/play/wwdc2020/10031>.
10. Dix, A.: Human-computer interaction, foundations and new paradigms. J. Vis. Lang. Comput. 42, 122–134 (2017)
11. Heer, J., Bostock, M.: Declarative language design for interactive visualization. IEEE Trans. Visual Comput. Graphics 16(6), 1149–1156 (2010)
12. Inie, N., Dalsgaard, P.: How interaction designers use tools to manage ideas. ACM Trans. Comput.-Hum. Interact. (TOCHI) 27(2), 1–26 (2020)
13. McGill, M., Kehoe, A., Freeman, E., Brewster, S.: Expanding the bounds of seated virtual workspaces. ACM Trans. Comput.-Hum. Interact. (TOCHI) 27(3), 1–40 (2020)
14. Microsoft 2020. Xamarin.Forms .NET. <https://dotnet.microsoft.com/apps/xamarin/xamarinforms>.
15. Tanenbaum, A., Wetheall, D.: Computer Networks. Pearson (2013)