1. **Паспорт Образовательной программы**

**«** Разработка мобильных приложений на платформе iOS **»**

|  |  |
| --- | --- |
| **Версия программы** | **1** |
| **Дата Версии** | 10**.**10**.**2020 |

1. **Сведения о Провайдере**

|  |  |  |
| --- | --- | --- |
| 1.1 | Провайдер | ФГБОУ ВО "Липецкий государственный технический университет" |
| 1.2 | Логотип образовательной организации | **логотип_QSG1nAq.jpg** |
| 1.3 | Провайдер ИНН | 4826012416 |
| 1.4 | Ответственный за программу ФИО | Корнеева Анастасия Олеговна |
| 1.5 | Ответственный должность | Специалист по учебно-методической работе деканата факультета дополнительного образования |
| 1.6 | Ответственный Телефон | +7(904)285-23-83 |
| 1.7 | Ответственный Е-mail | korneeva\_ao@stu.lipetsk.ru |

1. **Основные Данные**

|  |  |  |
| --- | --- | --- |
| **№** | **Название** | **Описание** |
| 2.1 | Название программы | Разработка мобильных приложений на платформе iOS |
| 2.2 | Ссылка на страницу программы | https://cloud.stu.lipetsk.ru/index.php/s/35YN2rWAM3xoZYd#pdfviewer |
| 2.3 | Формат обучения | Онлайн |
|  | Подтверждение от ОО наличия возможности реализации образовательной программы с применением электронного обучения и (или) дистанционных образовательных технологий с возможностью передачи данных в форме элементов цифрового следа | У каждого обучающегося будет заведен личный кабинет на сайте ЛГТУ.  http://eserv.stu.lipetsk.ru/course/view.php?id=672 |
| 2.4 | Уровень сложности | Продвинутый |
| 2.5 | Количество академических часов | **72** |
|  | Практикоориентированный характер образовательной программы: не менее 50 % трудоёмкости учебной деятельности отведено практическим занятиям и (или) выполнению практических заданий в режиме самостоятельной работы (кол-во академических часов) | 72% - 52 часа  36 часов практических занятий,  16 - самостоятельная работа по выполнению практических заданий |
| 2.6 | Стоимость обучения одного обучающегося по образовательной программе, а также предоставление ссылок на 3 (три) аналогичные образовательные программы иных организаций, осуществляющих обучение, для оценки объективности стоимости или обоснование уникальности представленной образовательной программы в случае отсутствия аналогичных образовательных программ на рынке образовательных услуг | 30000  1. Swift. Уровень 2. Объектно - ориентированное программирование – 32 часа, 29 990 руб.  https://www.specialist.ru/course/swift2  2. Разработка iOS приложений на Swift 5. Уровень 2 - 40 часов, 39 990 руб  https://www.specialist.ru/course/ajprog2  3. Онлайн-курс iOS-разработчик - 2 месяца, 28 000 руб.  https://contented.ru/edu/ios?utm\_source=infopartners&utm\_medium=partner&utm\_campaign=checkroi&utm\_term=regular&utm\_content=ios |
| 2.7 | Минимальное количество человек на курсе | 10 |
| 2.8 | Максимальное количество человек на курсе | 20 |
| 2.9 | Данные о количестве слушателей, ранее успешно прошедших обучение по образовательной программе | - |
| 2.10 | Формы аттестации | Итоговое тестирование, выполнение практического задания |
|  | Указание на область реализации компетенций цифровой экономики, к которой в большей степени относится образовательная программа, в соответствии с Перечнем областей | Программирование и создание ИТ-продуктов |

1. **Аннотация программы**

Наиболее полное и содержательное описание программы, которое включает:

1) общую характеристику компетенций, качественное изменение которых осуществляется в результате обучения или которые формируются в результате освоения образовательной программы;

2) описание требований и рекомендаций для обучения по образовательной программе;

3) краткое описание результатов обучения в свободной форме, а также описание востребованности результатов обучения в профессиональной деятельности.

Ограничение по размеру: не менее 1000 символов -?

Цели и задачи изучения

Целью реализации программы «Разработка мобильных приложений на платформе iOS» является изучение принципов функционирования аппаратно-программных средств современных мобильных устройств на платформе iOS, языка программирования Swift и технологий проектирования, реализации и сопровождения мобильных iOS приложений, а также развитие навыков разработки прикладного программного обеспечения для мобильных устройств.

Для достижения поставленной цели реализуются следующие задачи:

– изучение основ языка программирования Swift;

– изучение принципов проектирования и разработки пользовательского интерфейса для iOS приложения;

– изучения фреймворков для поддержки многопоточной работы, сетевого взаимодействия, работы с СУБД.

Требование к подготовленности обучающегося к освоению содержания учебной программы

Программа повышения квалификации «Разработка мобильных приложений на платформе iOS» предназначена для лиц, имеющих среднее профессиональное или высшее образование.

Результаты изучения программы

В результате освоения программы слушатель должен приобрести следующие компетенции:

способность планировать и разрабатывать архитектуру мобильного приложения на платформе iOS;

способность разрабатывать пользовательский интерфейс и основную логику для мобильного приложения на платформе iOS.

Знания, умения, навыки, приобретаемые обучающимися при изучении программы

В результате обучения обучающиеся должны:

знать:

особенности архитектуры платформы iOS;

современные инструменты разработки программного обеспечения для мобильных устройств;

жизненный цикл мобильного приложения;

основные элементы пользовательского интерфейса мобильных приложений;

структуру и основные компоненты интерфейсов прикладного программирования мобильных платформ;

способы организации многопоточной работы;

основные классы и методы фреймворков URLSession и Alamofire

основные классы и методы фреймворка CoreData

уметь:

разрабатывать пользовательский интерфейс iOS приложения в Storyboard;

выполнять отладку и тестирование мобильных приложений на физических устройствах и эмуляторах iOS;

планировать архитектуру iOS приложения;

выполнять проектирование и развертывание БД CoreData;

устанавливать расширения cocoapods.

владеть:

языком программирования Swift для платформы iOS;

навыками использования комплекта средств разработки iOS SDK и среды Xcode;

навыками оптимизации работы приложений для платформы iOS.

Краткое содержание дисциплины:

Содержание курса:

1) Основы синтаксиса языка Swift.

2) Объекто-ориентированное программирование в Swift.

3) Жизненный цикл iOS приложения.

4) Базовые элементы интерфейса

5) Стандартные виды View Controller.

6) Многопоточность

7) Работа с сетью

8) Работа с CoreData

1. ШАБЛОН ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ (ДПО)

Титульный лист программы

Название организации

Дополнительная профессиональная программа повышения квалификации

«Разработка мобильных приложений на платформе iOS »

72 час.

**ОБЩАЯ ХАРАКТЕРИСТИКА ПРОГРАММЫ**

**1.Цель программы**

Целью реализации программы «Разработка мобильных приложений на платформе iOS» является изучение принципов функционирования аппаратно-программных средств современных мобильных устройств на платформе iOS, языка программирования Swift и технологий проектирования, реализации и сопровождения мобильных iOS приложений, а также развитие навыков разработки прикладного программного обеспечения для мобильных устройств.

Для достижения поставленной цели реализуются следующие задачи:

– изучение основ языка программирования Swift;

– изучение принципов проектирования и разработки пользовательского интерфейса для iOS приложения;

– изучения фреймворков для поддержки многопоточной работы, сетевого взаимодействия, работы с СУБД.

**2.Планируемые результаты обучения:**

2.1.Знание (осведомленность в областях)

2.1.1. особенности архитектуры платформы iOS;

2.1.2. современные инструменты разработки программного обеспечения для мобильных устройств;

2.1.3. жизненный цикл мобильного приложения;

2.1.4. основные элементы пользовательского интерфейса мобильных приложений;

2.1.5. структуру и основные компоненты интерфейсов прикладного программирования мобильных платформ;

2.1.6. способы организации многопоточной работы;

2.1.7. основные классы и методы фреймворков URLSession и Alamofire

2.1.8. основные классы и методы фреймворка CoreData

2.2. Умение (способность к деятельности)

2.2.1. разрабатывать пользовательский интерфейс iOS приложения в Storyboard;

2.2.2. выполнять отладку и тестирование мобильных приложений на физических устройствах и эмуляторах iOS;

2.2.3. планировать архитектуру iOS приложения;

2.2.4. выполнять проектирование и развертывание БД CoreData;

2.2.5. устанавливать расширения cocoapods.

2.3.Навыки (использование конкретных инструментов)

2.3.1 использование языка программирования Swift для платформы iOS;

2.3.2. использования комплекта средств разработки iOS SDK и среды Xcode;

2.3.3. оптимизации работы приложений для платформы iOS.

**3.Категория слушателей** (возможно заполнение не всех полей)

* 1. Среднее профессиональное или высшее образование
  2. Наличие опыта профессиональной деятельности - приветствуется
  3. Рекомендуется: владение основами программирования на одном из языков высокого уровня (C++, C#, Java и т.д.), знание понятий и принципов объектно-ориентированного программирования.

**4.Учебный план программы «…..наименование программы….»**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ п/п** | **Модуль** | **Всего, час** | **Виды учебных занятий** | | |
| **лекции** | **практические занятия** | **самостоятельная работа** |
| 1 | Язык программирования Swift | 20 | 6 | 10 | 4 |
| 2 | Графический интерфейс мобильного приложения | 24 | 8 | 12 | 4 |
| 3 | Фреймворки и расширения cocoapods | 26 | 6 | 14 | 6 |
| **Итоговая аттестация** | |  | **Указывается вид (экзамен, зачёт, реферат и т.д.)** | | |
| Итоговый контроль | | 2 | Итоговое тестирование | | |

**5.Календарный план-график реализации образовательной** программы

(дата начала обучения – дата завершения обучения) в текущем календарном году, указания на периодичность набора групп (не менее 1 группы в месяц)

|  |  |  |  |
| --- | --- | --- | --- |
| **№ п/п** | **Наименование учебных модулей** | **Трудоёмкость (час)** | **Сроки обучения** |
| **1** | Язык программирования Swift | 20 | 09.11.2020 -11.11.2020 |
| **2** | Графический интерфейс мобильного приложения | 24 | 12.11.2020 - 16.11.2020 |
| 3 | Фреймворки и расширения cocoapods | 26 | 17.11.2020 - 21.11.2020 |
| 4 | Итоговая аттестация | 2 | 21.11.2020 |
| **Всего:** | | 72 | 09.11.2020 - 21.11.2020 |

**6.Учебно-тематический план программы «**  Разработка мобильных приложений на платформе iOS **»**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **№ п/п** | **Модуль / Тема** | **Всего, час** | **Виды учебных занятий** | | | **Формы контроля** |
| **лекции** | **практические занятия** | **самостоятельная работа** |
| 1 | Язык программирования Swift | 20 | 6 | 10 | 4 |  |
| 1.1 | Основы синтаксиса языка Swift | 8 | 2 | 4 | 2 | Контроль выполнения практических работ. |
| 1.2 | Объекто-ориентированное программирование в Swift | 12 | 4 | 6 | 2 | Контроль выполнения практических работ. |
| 2 | Графический интерфейс мобильного приложения | 24 | 8 | 12 | 4 |  |
| 2.1 | Жизненный цикл iOS приложения | 3 | 2 |  | 1 |  |
| 2.2 | Базовые элементы интерфейса | 7 | 2 | 4 | 1 | Контроль выполнения практических работ. |
| 2.3 | Стандартные виды View Controller | 14 | 4 | 8 | 2 | Контроль выполнения практических работ. |
| 3 | Фреймворки и расширения cocoapods | 26 | 6 | 14 | 6 |  |
| 3.1 | Многопоточность | 8 | 2 | 4 | 2 | Контроль выполнения практических работ. |
| 3.2 | Работа с сетью | 10 | 2 | 6 | 2 | Контроль выполнения практических работ. |
| 3.3 | Работа с CoreData | 8 | 2 | 4 | 2 | Контроль выполнения практических работ. |
| 4 | Итоговая аттестация | 2 |  |  | 2 | Итоговое тестирование |
|  |  |  |  |  |  |  |

**7. Учебная (рабочая) программа повышения квалификации «**  Разработка мобильных приложений на платформе iOS  **»**

**Модуль 1.** Язык программирования Swift **(**20 час**.)**

**Тема 1.1** Основы синтаксиса языка Swift **(**8 час.**)**

Основы синтаксиса языка Swift. Константы и переменные. Числовые типы. Безопасность и определение типов. Конвертация типов. Логический тип. Кортежи. Опциональные значения. Базовые операторы. Операторы диапазона. Логические операторы. Строки и символы. Массивы. Словари. Циклы. Условные выражения. Обработка ошибок.

**Тема 1.2** Объекто-ориентированное программирование в Swift (12 час.)

Функции, замыкания. Параметры функций и возврат значений. Опциональный кортежи, как возвращаемый тип. Тип-функция. Вложенные функции. Замыкания. Trailing closures. Захват значений замыканиями.

Классы, Структуры, Перечисления. Хранимые свойства. Ленивые свойства. Вычисляемые свойства. Обозреватели свойств. Свойства типов. Имена параметров методов. Mutating методы. Методы типов. Индексаторы. Наследование. Переопределение. Инициализация. Дефолтные инициализаторы. Memberwise-инициализаторы. Делегация инициализации. Наследование и инициализация. Failable-инициализаторы. Required-инициализаторы. Деинициализация.

**Модуль 2.** Графический интерфейс мобильного приложения (24 час.)

**Тема 2.1.** Жизненный цикл iOS приложения (3 час.)

Жизненный цикл iOS приложения. Состояния приложения. Жизненный цикл View Controller. Модель MVC. Модель MVVM.

Тема 2.2. Базовые элементы интерфейса (7 час.)

Верстка экранов в Storyboard. Ручное создание View. Background. Кнопки. Метки. Constraints – ограничения по расстоянию, по размещению, размерам. Привязки. Stack View. Масштабирование. Привязка графических элементов к членам класса. Обработчики событий. UIImageView. UIPickerView. Делегирование и источник данных. Сохранение пользовательской информации (UserDefaults). Текстовые поля TextField. AlertController. Работа с последовательностью View. NavigationController. Работа с сигвеями (Segues). Ручной вызов сигвея. Передача данных в Segue и обратно.

Тема 2.3. Стандартные виды View Controller (14 час)

Table View Controller. Контролер и элемент TableView. UITableViewCell. Дизайн ячейки. Протоколы Delegate и DataSource. Добавление и удаление данных из TableView. Передача данных. Сохранение данных

CollectionViewController. Принцип работы CollectionView. Произвольные ячейки. UICollectionViewCell. Работа с градиентными слоями. Использование нескольких секций. Добавление элементов в CollectionView. Обновление контента. Формирование произвольных ячеек с помощью ViewFlowLayout. Удаление ячеек

Модуль 3. Фреймворки и расширения cocoapods (26 час.)

Тема 3.1. Многопоточность (8 час.)

Grand Central Dispatch. Operation. Последовательные и параллельные очереди. Синхронное и асинхронное выполнение заданий. Глобальные очереди. Проблемы многопоточности. Паттерны для типовых задач

Тема 3.2. Работа с сетью (10 час.)

URLSession. Cocoapods (менеджер зависимостей). Установка Cocoapods. Podfile. Библиотека Alamofire.

Тема 3.3. Работа с CoreData (8 час)

Стек технологий Core Data. Управляемая объектная модель. Координатор постоянного хранилища. Контексты управляемого объекта

**Описание практико-ориентированных заданий и кейсов**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Номер темы/модуля** | **Наименование практического занятия** | **Описание** |
| 1. | Модуль 1. Язык программирования Swift |  |  |
| 1.1 | Тема 1.1. Основы синтаксиса языка Swift | Основы работы в playground | Конвертация типов. Логический тип. Кортежи. Опциональные значения. Базовые операторы. Операторы диапазона. Логические операторы. Строки и символы. Массивы. Словари. Циклы. Условные выражения. Обработка ошибок. |
|  |  | Разработка консольного приложения | Разработка консольного приложения на языке Swift – сортировка и поиск данных |
| 1.2. | Тема 1.2 Объекто-ориентированное программирование в Swift | Функции и замыкания | Работа с функциями и замыканиями. Параметры функций и возврат значений. Тип-функция. Вложенные функции. Замыкания. Trailing closures  Захват значений замыканиями. |
|  |  | Классы и структуры | Хранимые свойства. Ленивые свойства. Вычисляемые свойства. Обозреватели свойств.  Свойства типов. Имена параметров методов. Mutating методы. Методы типов. Индексаторы.  ARC. Использование weak и unowned. Опциональное связывание. Проверка типа. Downcasting и upcasting. Any и AnyObject. Вложенные типы. Типы доступа.  Протоколы, расширения  Использование расщирений. Протоколы. Реализация протоколов. |
|  |  | Разработка консольного объектно-ориентированного приложения на языке Swift – игра | Разработка консольного объектно-ориентированного приложения на языке Swift – игру-ходилка с кубиком и фишками для 2-4 игроков |
| 2. | Модуль 2. Графический интерфейс мобильного приложения |  |  |
| 2.1 | Тема 2.2. Базовые элементы интерфейса | Дизайн экрана приложения в Storyboard. | Создание ViewController.  Добавление графических элементов.  Верстка с использованием констреймов и AutoLayout.  Привязка графических элементов к членам класса.  Реализация обработчиков событий |
|  |  | Ручная верстка экрана | Программное создание ViewController.  Программное добавление графических элементов и обработчкиов событий.  Программное добавление констреймов. |
|  | Тема 2.3. Стандартные виды View Controller | Работа с TableViewController | Создание контролера TableViewController.  Дизайн ячейки UITableViewCell.  Реализация методов протоколов Delegate и DataSource.  Добавление и удаление данных из TableView.  Реализация просмотра содержимого ячейки |
|  |  | Работа с CollectionViewController | Создание контроллера CollectionViewController  Дизайн ячейки. UICollectionViewCell.  Реализация нескольких секций.  Добавление и удаление элементов из CollectionView.  Реализация просмотра содержимого ячейки |
|  |  | Работа с распознаванием жестов | Механизм распознания жестов. UIGestureRecognizer. Типовые жесты: одиночный тап, панорамирование (перетаскивание), щипки, вращение. |
|  |  | Работа со Scroll View | Работа с Zoom. Вложенный Scroll View. Работа с клавиатурой. |
|  | Модуль 3. Фреймворки и расширения cocoapods |  |  |
|  | Тема 3.1. Многопоточность |  | Применение синхронного и асинхронного выполнение заданий для реализации бекенд задач |
|  |  |  | Применение глобальных очередей для решения бекенд задач |
|  | Тема 3.2. Работа с сетью | Работа с URLSession | Выполнение HTTP-запросов с помощью фреймворка URLSession |
|  |  | Работа с cocoapods | Cocoapods (менеджер зависимостей) Установка Cocoapods. Редактирование Podfile  Установка библиотеки Alamofire |
|  |  | Работа с Alamofire | Выполнение HTTP-запросов с помощью фреймворка Alamofire |
|  | Тема 3.3. Работа с CoreData | Разработка структуры БД в CoreData | Планирование структуры БД в CoreData. Создание БД CoreData |
|  |  | Работа с CoreData | Работа с БД с помощью фреймворка Core Data |
|  |  |  |  |

**8.Оценочные материалы по образовательной программе**

**8.1. Вопросы тестирования по модулям**

|  |  |  |  |
| --- | --- | --- | --- |
| **№ модуля** | **Вопросы входного тестирования** | **Вопросы промежуточного тестирования** | **Вопросы итогового тестирования** |
| **1.1** | 1. Дан массив из 20 элементов. На любом языке программирования напишите процедуру нахождение суммы элементов этого массива  2. Дан массив из 20 элементов. На любом языке программирования напишите процедуру сортировки элементов данного массива по возрастанию  3. Дан упорядоченный массив из 20 элементов. На любом языке программирования напишите процедуру ускоренного поиска элемента в данном массиве  4. На любом языке программирования напишите объявление класса "Студент", пригодного для хранения информации об обучающихся в университете  5. На любом языке программирования напишите иерархию классов (базовый и 2 потомка), описывающих некоторых работников с почасовой оплатой (один из потомков) и фиксированной оплатой (второй потомок). Предусмотреть метод для расчета среднемесячной заработной платы. | 1. Константы и переменные. Числовые типы.  2. Безопасность и определение типов.  3. Опциональные значения.  4. Функции. Вложенные функции.  5. Замыкания. Trailing closures. Захват значений замыканиями.  6. Классы. Наследование.  7. Инициализация. Деинициализация  8. Расширения классов.  9. Протоколы | 1. Константы и переменные. Числовые типы.  2. Безопасность и определение типов.  3. Опциональные значения.  4. Функции. Вложенные функции.  5. Замыкания. Trailing closures. Захват значений замыканиями.  6. Классы. Наследование.  7. Инициализация. Деинициализация  8. Расширения классов.  9. Протоколы |
| 1.2 |  | 1. Жизненный цикл iOS приложения  2. Жизненный цикл View Controller  3. Типовые элементы графического интерфейса.  4. Делегирование и источник данных  5. NavigationController.  6. Работа с сигвеями (Segues). Передача данных в Segue и обратно.  7. Table View Controller  8. CollectionViewController  9. Scroll View  10. Механизм распознания жестов. Типовые жесты: одиночный тап, панорамирование (перетаскивание), щипки, вращение | 1. Жизненный цикл iOS приложения  2. Жизненный цикл View Controller  3. Типовые элементы графического интерфейса.  4. Делегирование и источник данных  5. NavigationController.  6. Работа с сигвеями (Segues). Передача данных в Segue и обратно.  7. Table View Controller  8. CollectionViewController  9. Scroll View  10. Механизм распознания жестов. Типовые жесты: одиночный тап, панорамирование (перетаскивание), щипки, вращение |
| 1.3 |  | 1. Фреймворк URLSession  2. Фреймворк Alamofire  3. Последовательные и параллельные очереди.  4. Глобальные очереди  5. Синхронное и асинхронное выполнение заданий  6. Стек технологий Core Data  7. Управляемая объектная модель  8. Координатор постоянного хранилища | 1. Фреймворк URLSession  2. Фреймворк Alamofire  3. Последовательные и параллельные очереди.  4. Глобальные очереди  5. Синхронное и асинхронное выполнение заданий  6. Стек технологий Core Data  7. Управляемая объектная модель  8. Координатор постоянного хранилища |

**8.2.** Входной контроль охватывает всех обучаемых и проводится в форме собеседования. Целью его является определение уровня знаний обучаемых в области программирования. Каждый обучающийся должен написать ответ-эссе (10-20 строк кода на любом языке программирования) на следующие вопросы:

1. Дан массив из 20 элементов. На любом языке программирования напишите процедуру нахождение суммы элементов этого массива

2. Дан массив из 20 элементов. На любом языке программирования напишите процедуру сортировки элементов данного массива по возрастанию

3. Дан упорядоченный массив из 20 элементов. На любом языке программирования напишите процедуру ускоренного поиска элемента в данном массиве

4. На любом языке программирования напишите объявление класса "Студент", пригодного для хранения информации об обучающихся в университете

5. На любом языке программирования напишите иерархию классов (базовый и 2 потомка), описывающих некоторых работников с почасовой оплатой (один из потомков) и фиксированной оплатой (второй потомок). Предусмотреть метод для расчета среднемесячной заработной платы.

Итоговая аттестация проводится в формате тестирования.

Пример тестовых вопросов для итогового тестирования:

1. Для чего нужны автозамыкания

А) Для того что бы передавать выражение в качестве аргумента функции

Б) такого нет в swift

В) для того чтобы сохранять значение переменных внутри замыкания

2. Возникнет ли ошибка при выполнении следующего кода:

var num: Int?

print(num ?? 0)

А) Да, возникнет

Б) Нет, не возникнет

3. В каком случае лучше использовать структуру, а не класс?

А) Когда нам нужно инкапсулировать несколько связанных ссылочных типов

Б) Лучше всегда использовать класс

В) Особой разницы нет, можно всегда использовать структуру вместо класса

Г) Когда нам нужно инкапсулировать несколько связанных типов значения

4. Ниже описаны структура и класс с одинаковым набором полей:

struct someStruct {

var x = 5

}

class someClass {

var x = 5

}

В каком варианте верно показана инициализация для класса и структуры?

1. let someStructObj = simeStruct()

2. let someStructObj = someStruct(x: 10)

3. let someClassObj = someClass()

4. let someClassObj = someClass(x: 10)

А) 1, 2, 3

Б) 1, 3, 4

В) 1, 2, 3, 4

Г) 1, 3

5. Что означает ключевое слово @escaping?

A) то что замыкание не обязательно вызывать внутри функции

Б) ничего, такого нет в swift

В) то что замыкание "сбегающее" и может быть вызвано уже после того

6. Какого типа будет переменная:

var x = 10.0

А) Float

Б) String

В) Double

Г) Int

7. Что произойдет при выполнении кода:

let x = 10.3

let y : Float = 20.7

let z = 9

let sum = z + y + x

print(sum)

А) Будет ошибка, так не сделано приведение типов

Б) В консоль распечатается число 40

8. Можно ли через расширение добавить поддержку того или иного протокола?

А) нет, только добавить методы

Б) да

9. Можно ли через расширение добавить поддержку того или иного протокола?

А) нет, только добавить методы

Б) да

10. Что выведет в консоль данный код:

let text = ["hello", "world", "!"]

for index in 0... text.count {

print(strings[index])

}

А) "hello", "world", "!"

Б) возникнет ошибка

11. Выберите все операнды, которые можно использовать в условии if

А) ==, !=, <=, >=, <>, &&, ||

Б) ==, !=, <=, >=, <>

В) <=, >=, <>

Г) ==, !=, <=, >=, &&, ||

Д) ==, !=, <=, >=, <>, &&, ||, &, |

12. Ниже дана функция, принимающая параметром замыкание, на каком варианте показан вызов этой функции?

1. someFunc(closure: {"тело замыкания"})

2. someFunc() {"тело замыкания"}

func sumeFunc(closure : () -> Void) {

"тело функции"

}

А) только №1

Б) неверны оба

В) верны оба

Г) только №2

13. Как будет выглядеть вызов функции ниже:

func function(\_ x0: Int, x1: Int, x2: Int = 30) {

print(x0 + x1 + x2)

}

А) это неверное объявление функции

Б) только function(1, x1: 2, x2: 3)

В) function(1, x1: 2, x2: 3) или function(1, x1: 2)

Г) function(x0: 1, x1: 2, x2: 3) или function(x0: 1, x1: 2)

14. При определении метода как шаблонного можно указать только один шаблонный тип?

А) да, только один

Б) нет, функция может использовать одновременно несколько шаблонных типов

15. Написать класс Банкомат. Методы: Выдать, Принять. Свойство: Наличность. Если денег не хватает, то генерировать ошибку. Если при приеме денег в пачке больше 100 000, то тоже генерировать ошибку и соответственно их обрабатывать.

Класс Персона. Есть ссылка на кредитную карту. Свойство: Наличность.

Класс Кредитная карта. Свойство: Безнал. Есть ссылка на владельца.

У персоны может и не быть кредитной карты. При попытке снять или положить деньги должно выводиться сообщение.

Необходимо сделать проверку достаточно ли средств на кредитной карте.

При обнулении (=nil) клиента, его кредитная карта должна освобождать память.

16. Написать классы Vector2D (x, y) и Vector3D (x, y, z)

Написать протокол c 2-мя методами:

  vectorLength -> Double,

  printComponets() -> Void

Подписать классы протоколом

Создать массив из смешанных векторов

В цикле вывести компоненты вектора и его длину

Текущий контроль освоения дисциплины осуществляется на основании контроля выполнения практических работ:

Практическая работа №1 “ Разработка консольного приложения на языке Swift ”;

Практическая работа №2 “ Разработка консольной игры на языке Swift ”;

Практическая работа №3 “Дизайн экрана приложения в Storyboard”;

Практическая работа №4 “Ручная верстка экрана приложения”;

Практическая работа №5 “Разработка приложения на основе TableViewController”;

Практическая работа №6 “Разработка приложения на основе CollectionViewController”;

Практическая работа №7 “Выполнение HTTP-запросов с помощью фреймворка URLSession”;

Практическая работа №8 “Работа с БД с помощью фреймворка Core Data”;

Выполнение практических работ заключается в написании программного продукта, соответствующего требованиям, указанным в методических указаниях. Выполнение практической работы осуществляется каждым студентом самостоятельно. Все неясные моменты относительно выполнения практической работы нужно выяснять у преподавателя.

Текущий контроль проводится на практических занятиях с целью определения качества усвоения учебного материала дисциплины, включая модули, предназначенные для самостоятельного изучения. При этом контролируется адекватное использование специальной технической терминологии, знание основных понятий, моделей, методов и процедур, успешное применение подходящих методов и алгоритмов, актуальных стандартов и популярных сторонних библиотек

Критерии оценки практических работ:

- структура и логика работы (соответствие предложенному плану)

- знание основных понятий, моделей, методов и процедур

- успешное применение подходящих методов и алгоритмов

- качество и оформление программного кода

**.**

**8.3.** Примеры контрольных вопросов

Модуль 1. Язык программирования Swift

- переменные и константы

- опциональные значения; извлечение опционального значения

- кортежи

- диапазоны

- массивы

- словари

- управляющие конструкции

- замыкания

- перечисления

- структуры

- классы: методы и переменные экземпляра, методы и переменные класса

- классы: ленивые свойства, наблюдатели свойств, сабскрипты, наследование

- инициализация классов; виды инициализаторов; делегирование инициализации, двухфазная инициализация

- опциональная последовательность

- расширения

- протоколы

- ARC – автоматический подсчет ссылок: сильные ссылки, слабые ссылки, безхозные ссылки

- паттерн «data source»

- паттерн «delegate»..

Модуль 2. Графический интерфейс мобильного приложения

- использование констреймов

- использование привязок

- связывание элементов интерфейса с полями членами класса

- автомасштабирование

- UIStackView

- класс UIView

- класс UIViewController

- программное создание элементов интерфейса

- программное создание констреймов

- класс UITableView

- протокол UITableViewDataSource

- протокол UITableViewDelegate

- шаблоны ячеек

- добавление и удаление данных из TableView

- класс UICollectionView

- протокол UICollectionViewDataSource

- протокол UICollectionViewDelegate

- шаблоны ячеек

- добавление и удаление данных из CollectionView

Модуль 3. Фреймворки и расширения cocoapods

- стандартная сессия для типовых задач

- конфигурирование объекта сессии с помощью URLSessionConfiguration

- виды задач URLSessionTask

- синхронное выполнение задач

- асинхронное выполнение задач

- последовательные и параллельные очереди

- глобальные очереди

- cтек технологий Core Data

- управляемая объектная модель

- использование координатора постоянного хранилища

- контекст управляемого объекта

**.**

**8.4.**  Примеры обучающих задач (кейсов)

1. Практическая работа №1 . Разработка консольного приложения на языке Swift

Задание: написать консольное приложение, предназначенное для сортировки массива кортежей (Int, Float, String) и поиска данных по массиву. Метод сортировки и поиска приведен в таблице №1. Массив может быть отсортирован по полю Int или по полю Float; поле, по которому осуществляется сортировка и поиск, выбирается пользователем

Программа должна выполнять следующее:

- Запрос размера массива у пользователя и заполнение массива случайными данными

- Выбор поля, по которому выполняется сортировка

- Сортировка массива по заданному полю заданным по варианту методом

- Поиск в упорядоченном массиве значений заданным по варианту методом

- Сортировка и поиск должны выполняться в отдельных функциях

- Передача информации о выбранном поле для поиска и сортировки должно выполняться с помощью замыкания

2. Практическая работа №2. Разработка консольной игры на языке Swift

Задание: написать консольную игру-ходилку с кубиком и фишками для 2-4 игроков; параметры игры и типы используемых секретных клеток приведены в таблице №1.

Программа должна выполнять следующее:

- Генерация поля игры заданного размера (использовать паттерн «data source»)

- Выбор количества игроков

- Текстовое отображение процесса игры (использовать паттерн «delegate»)

3. Практическая работа №3. Дизайн экрана приложения в Storyboard

Задание: сверстать заданный экран приложения в Storyboard с использованием autolayout констрейнтов.

4. Практическая работа №4. Ручная верстка экрана приложения

Задание: сверстать заданный экран приложения программным образом с использованием autolayout констрейнтов.

**.**

**8.5.**  Итоговая оценка выставляется на основании итогового тестирования (20 вопросов с вариантами ответов):

1-9 правильных ответов - 2 баллов - "неудовлетворительно", не зачтено

10-14 правильных ответов - 3 балла - "удовлетворительно", зачтено

15-18 правильных ответов - 4 балла - "хорошо", зачтено

19-20 правильных ответов - 5 баллов - "отлично", зачтено **.**

**9.Организационно-педагогические условия реализации программы**

**9.1. Кадровое обеспечение программы**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№**  **п/п** | **Фамилия, имя, отчество (при наличии)** | **Место основной работы и должность, ученая степень и ученое звание (при наличии)** | **Ссылки на веб-страницы с портфолио (при наличии)** | **Фото в формате jpeg** | **Отметка о полученном согласии на обработку персональных данных** |
| **1** | Домашнев Павел Алексеевич | ФГБОУ ВО "Липецкий государственный технический университет",  доцент кафедры Автоматизированных систем управления,  к.т.н. | <http://www.stu.lipetsk.ru/>  struct/kaf/asu/about/staff.html | D:\Foto\Павел\P1010977.JPG |  |

**9.2.Учебно-методическое обеспечение и информационное сопровождение**

|  |  |
| --- | --- |
| **Учебно-методические материалы** | |
| Методы, формы и технологии | Методические разработки,  материалы курса, учебная литература |
| Лекционные и практические занятия | The swift programming language <https://docs.swift.org/swift-book/> |
| Лекционные и практические занятия | Apple Developer Documentation <https://developer.apple.com/documentation> |

|  |  |
| --- | --- |
| **Информационное сопровождение** | |
| Электронные  образовательные ресурсы | Электронные  информационные ресурсы |
| Соколова В.В. Разработка мобильных приложений [Электронный ресурс]: учебное пособие/ Соколова В.В. — Электрон. текстовые данные. — Томск: Томский политехнический университет, 2014. — 176 c. — Режим доступа: http://www.iprbookshop.ru/34706.html. | ЭБС IPRbooks |
| Верескун Д.М. Разработка мобильных приложений для бизнеса [Электронный ресурс]: учебное пособие / Верескун Д.М. — Электрон. текстовые данные. — Саратов: Саратовский государственный технический университет имени Ю.А. Гагарина, ЭБС АСВ, 2012. — 51 c. — Режим доступа: http://www.iprbookshop.ru/76508.html. — ЭБС «IPRbooks» | ЭБС IPRbooks |
| Федотенко, М.А. Разработка мобильных приложений. Первые шаги [Электронный ресурс] / ред.: В.В. Тарапата, М.А. Федотенко. — Эл. изд. — М. : Лаборатория знаний, 2019. — 338 с. : ил. — (Школа юного программиста). — Деривативное эл. изд. на основе печ. аналога (М.: Лаборатория знаний, 2019); Электрон. текстовые дан. (1 файл pdf : 338 с.); Систем. требования: Adobe Reader XI; экран 10". — ISBN 978-5-00101-640-3. — Режим доступа: https://rucont.ru/efd/683979 | ЭБС Руконт "Контекстум" |

**9.3.Материально-технические условия реализации программы**

|  |  |
| --- | --- |
| Вид занятий | Наименование оборудования,  программного обеспечения |
| Лекционные занятия | Компьютер/ноутбук, доступ к сети Интернет, доступ в систему Moodle |
| Практические занятия | MacBook, среда разработки Xcode |

**III.Паспорт компетенций (Приложение 2)**

Описание перечня профессиональных компетенций, качественное изменение которых осуществляется в результате обучения.

Планируемые результаты обучения должны быть определены в виде знаний, умений, навыков и (или) опыта деятельности, характеризующих формирование/развитие компетенции(-й) в области цифровой экономики и представлены в виде Паспорта компетенций в машиночитаемом текстовом формате. Структура паспорта представлена в приложении.

ПАСПОРТ КОМПЕТЕНЦИИ

(Разработка мобильных приложений на платформе iOS)

(ФГБОУ ВО "Липецкий государственный технический университет" )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1. | Наименование компетенции | | Способность планировать и разрабатывать архитектуру, пользовательский интерфейс и основную логику мобильного приложения на платформе iOS | |
| 2. | Указание типа компетенции |  | Профессиональная | |
| 3. | Определение, содержание и основные сущностные характеристики компетенции | | Знает:  особенности архитектуры платформы iOS;  современные инструменты разработки программного обеспечения для мобильных устройств;  жизненный цикл мобильного приложения;  основные элементы пользовательского интерфейса мобильных приложений;  структуру и основные компоненты интерфейсов прикладного программирования мобильных платформ;  способы организации многопоточной работы;  основные классы и методы фреймворков URLSession и Alamofire  основные классы и методы фреймворка CoreData  Умеет:  разрабатывать пользовательский интерфейс iOS приложения в Storyboard;  выполнять отладку и тестирование мобильных приложений на физических устройствах и эмуляторах iOS;  планировать архитектуру iOS приложения;  выполнять проектирование и развертывание БД CoreData;  устанавливать расширения cocoapods.  Владеет:  языком программирования Swift для платформы iOS;  навыками использования комплекта средств разработки iOS SDK и среды Xcode;  навыками оптимизации работы приложений для платформы iOS. | |
| 4. | Дескриптор знаний, умений и навыков по уровням | | Уровни сформированности компетенции  обучающегося | Индикаторы |
|  | | Начальный уровень  (Компетенция недостаточно развита. Частично проявляет навыки, входящие в состав компетенции. Пытается, стремится проявлять нужные навыки, понимает их необходимость, но у него не всегда получается.) | Компетенция недостаточно развита.  Частично проявляет навыки владения языком программирования Swift для платформы iOS.  Умеет разрабатывать пользовательский интерфейс iOS приложения в Storyboard.  Знает основные элементы пользовательского интерфейса мобильных приложений, структуру и основные компоненты интерфейсов прикладного программирования мобильных платформ |
|  | | Базовый уровень  (Уверенно владеет навыками, способен, проявлять соответствующие навыки в ситуациях с элементами неопределён-ности, сложности.) | Уверенно владеет языком программирования Swift для платформы iOS и навыками использования комплекта средств разработки iOS SDK и среды Xcode.  Умеет разрабатывать пользовательский интерфейс iOS приложения в Storyboard, выполнять отладку и тестирование мобильных приложений на физических устройствах и эмуляторах iOS, устанавливать расширения cocoapods  Знает особенности архитектуры платформы iOS, современные инструменты разработки программного обеспечения для мобильных устройств, жизненный цикл мобильного приложения, структуру и компоненты интерфейсов прикладного программирования мобильных платформ |
|  | | Продвинутый  (Владеет сложными навыками, способен активно влиять на происходящее, проявлять соответствующие навыки в ситуациях повышенной сложности.) | Владеет языком программирования Swift для платформы iOS и навыками использования комплекта средств разработки iOS SDK и среды Xcode. навыками оптимизации работы приложений для платформы iOS  Умеет разрабатывать продвинутый пользовательский интерфейс iOS приложения, выполнять отладку и тестирование мобильных приложений на физических устройствах и эмуляторах iOS, планировать архитектуру iOS приложения и выполнять проектирование и развертывание БД CoreData  Знает особенности архитектуры платформы iOS и жизненного цикла мобильного приложения,  структуру и основные компоненты интерфейсов прикладного программирования мобильных платформ, способы организации многопоточной работы, основные классы и методы фреймворков URLSession и Alamofire, основные классы и методы фреймворка CoreData |
| 5. | Характеристика взаимосвязи данной компетенции с другими компетенциями/ необходимость владения другими компетенциями для формирования данной компетенции | | Владение основами программирования на одном из языков высокого уровня (C++, C#, Java и т.д.)  Знание понятий и принципов объектно-ориентированного программирования. | |
| 6. | Средства и технологии оценки | | 1. Контроль выполнения практических работ – оценка оформленных отчетов по результатам выполнения практических работ  2. Итоговое тестирование | |

**VI.Иная информация о качестве и востребованности образовательной программы** (результаты профессионально-общественной аккредитации образовательной программы, включение в системы рейтингования, призовые места по результатам проведения конкурсов образовательных программ и др.) (при наличии)

**V.Рекомендаций к программе от работодателей**: наличие не менее двух писем и/или подтверждения на цифровой платформе Государственной системы предоставления ПЦС от работодателей о рекомендации образовательной программы для реализации в рамках Государственной системы предоставления ПЦС на формирование у трудоспособного населения компетенций цифровой экономики с указанием востребованности результатов освоения программы в сфере деятельности соответствующих компаний и готовности к рассмотрению заявок наиболее успешно освоивших образовательную программу граждан на прохождение стажировки и (или) собеседования на предмет трудоустройства путем проставления отметки в профиле программы

1. ООО «МедСофт»,

2. ОБУ «Информационно-технический центр»

3. Управление информатизации администрации Липецкой области,

4. ООО «МедСофт»,

5. Филиал ОАО «РЖД».

**VI.Указание на возможные сценарии профессиональной траектории граждан** по итогам освоения образовательной программы (в соответствии с приложением)

1. Трудоустройство

2. Развитие компетенций в текущей сфере занятости с целью развития профессиональных качеств, повышения уровня дохода

3. Переход в новую сферу занятости с целью повышение уровня дохода, расширение профессиональной деятельности

**VII.Дополнительная информация**

1. Swift. Уровень 2. Объектно - ориентированное программирование – 32 часа.

https://www.specialist.ru/course/swift2

2. Разработка iOS приложений на Swift 5. Уровень 1 - 24 часа

https://www.specialist.ru/course/iphone1

3. Разработка iOS приложений на Swift 5. Уровень 2 - 40 часов

https://www.specialist.ru/course/ajprog2

4. Онлайн-курс iOS-разработчик - 2 месяца.

https://contented.ru/edu/ios?utm\_source=infopartners&utm\_medium=partner&utm\_campaign=checkroi&utm\_term=regular&utm\_content=ios

5. iOS разработчик с нуля - 24 онлайн урока

https://swiftlab.ru/iOS-developer/?utm\_source=checkroi.ru&utm\_medium=checkroi.ru&utm\_campaign=checkroi.ru

**VIII.Приложенные Скан-копии**

Утвержденной рабочей программа (подпись, печать, в формате pdf)