

**Образовательное частное учреждение
Дополнительного профессионального образования «Центр
компьютерного обучения «Специалист» Учебно-научного центра при
МГТУ им. Н.Э. Баумана
(ОЧУ «Специалист»)**

123317 Москва, Пресненская набережная, д. 8, стр. 1, этаж 48, помещение 484с, комната 5,
ИНН 7701257303, ОГРН 1037739408189

Утверждаю:

Директор ОЧУ «Специалист»



/Т.С. Григорьева/
«14» февраля 2018 года

**Дополнительная профессиональная программа
повышения квалификации**

«SWIFT. Основы программирования.»

город Москва

Программа разработана в соответствии с Федеральным законом от 29 декабря 2012 г. №273-ФЗ «Об образовании в Российской Федерации», приказом Министерства образования и науки Российской Федерации от 1 июля 2013 г. N 499 "Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным профессиональным программам", Профессиональным стандартом «Программист» от 12.12.2016 №727н и на основании преемственности по отношению к Федеральному государственному образовательному стандарту высшего образования (ФГОС ВО) по направлениям подготовки 09.03.02 «Информационные системы и технологии» от 10.03.2015 №36623

Повышение квалификации слушателей, осуществляемое в соответствии с программой, проводится с использованием модульного принципа построения учебного плана с применением различных образовательных технологий, в том числе дистанционных образовательных технологий и электронного обучения в соответствии с законодательством об образовании.

Дополнительная профессиональная программа повышения квалификации, разработана образовательной организацией в соответствии с законодательством Российской Федерации, включает все модули, указанные в учебном плане.

Содержание оценочных и методических материалов определяется образовательной организацией самостоятельно с учетом положений законодательства об образовании Российской Федерации.

Структура дополнительной профессиональной программы соответствует требованиям Порядка организации и осуществления образовательной деятельности по дополнительным профессиональным программам, утвержденного приказом Минобрнауки России от 1 июля 2013 г. N 499.

Объем дополнительной профессиональной программы вне зависимости от применяемых образовательных технологий, должен быть не менее 16 академических часов. Сроки ее освоения определяются образовательной организацией самостоятельно.

Формы обучения слушателей (очная, очно-заочная, заочная) определяются образовательной организацией самостоятельно.

К освоению дополнительных профессиональных программ допускаются:

- лица, имеющие среднее профессиональное и (или) высшее образование;
- лица, получающие среднее профессиональное и (или) высшее образование.

Для определения структуры дополнительной профессиональной программы и трудоемкости ее освоения может применяться система зачетных единиц. Количество зачетных единиц по дополнительной профессиональной программе устанавливается организацией.

Образовательная деятельность слушателей предусматривает следующие виды учебных занятий и учебных работ: лекции, практические и семинарские занятия, лабораторные работы, круглые столы, мастер-классы, мастерские, деловые игры, ролевые игры, тренинги, семинары по обмену опытом, выездные занятия, консультации, выполнение аттестационной, дипломной, проектной работы и другие виды учебных занятий и учебных работ, определенные учебным планом.

1. Цель программы:

Основной целью курса является – получение слушателями навыков работы в среде программирования; реализации построенных алгоритмов в виде программы на конкретном языке программирования, изучение основ программирования на языке Swift и разработки информационных технологий для MacOS с целью решения практических задач.

Планируемый результат обучения:

Лица, успешно освоившие программу, будут владеть методами проектирования базовых и прикладных технологий для решения профессиональных задач для MacOS; средствами XCode для реализации информационных технологий для MacOS; способами формирования новых конкурентоспособных идей и реализации их средствами XCode

Совершенствуемые компетенции

№	Компетенция	Направление подготовки
		ФЕДЕРАЛЬНЫЙ ГОСУДАРСТВЕННЫЙ ОБРАЗОВАТЕЛЬНЫЙ СТАНДАРТ ВО НАПРАВЛЕНИЮ ПОДГОТОВКИ 09.03.02 ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ (уровень бакалавриата)
		Код компетенции
1	Способность к проектированию базовых и прикладных информационных технологий	ПК-11
2	Способность разрабатывать средства реализации информационных технологий (методические, информационные, математические, алгоритмические, технические и программные)	ПК-12
3	Способность формировать новые конкурентоспособные идеи и реализовывать их в проектах	ПК-27

Совершенствуемые компетенции в соответствии с трудовыми функциями профессионального стандарта «Программист»

№	Компетенция	Направление подготовки
		Трудовые функции (код)
1	Разработка и отладка программного кода	A/01.3, A/02.3, A/03.3, A/04.3, A/05.3
2	Проверка работоспособности и рефакторинг кода программного обеспечения	B/01.4, B/02.4, B/03.4, B/04.4
3	Интеграция программных модулей и компонент и верификация выпусков программного продукта	C/01.5, C/02.5
4	Разработка требований и проектирование программного обеспечения	D/01.6, D/02.6, D/03.6

После окончания обучения Слушатель будет знать:

- этапы решения задачи на компьютере;
- типы данных;
- базовые конструкции изучаемых языков программирования;
- принципы структурного и модульного программирования;

- принципы объектно-ориентированного программирования.

После окончания обучения Слушатель будет уметь:

- Применять методы проектирования технологий для решения прикладных задач
- Применять средства XCode для реализации информационных технологий
- Использовать методы формирования новых конкурентоспособных идей и реализации их средствами XCode
- Работать с кортежами
- Работать с опционалами
- Использовать преимущества конструкции switch в Swift
- Использовать оператор guard
- Работать с массивами, словарями и наборами
- Использовать замыкания

Данный курс соответствует требованиям профессионального стандарта «Программист»

2. Учебный план:

Категория слушателей: новички в программировании.

Требования к предварительной подготовке: «Основы программирования и баз данных», «Mac OS X: Работа с Mac для пользователей», «Английский язык для IT специалистов (elementary)».

Для прохождения курса в центре вам потребуется собственный Macbook с версией операционной системы не ниже MacOS 10.12.6 Sierra и установленным XCode 9

.Срок обучения: 24 академических часа, самостоятельных занятий 12 академических часов.

Форма обучения: очная, очно-заочная, заочная. По желанию слушателя форма обучения может быть изменена и/или дополнена.

Режим занятий: дневной, вечерний, группы выходного дня.

№ п/п	Наименование модулей по программе	Общая трудоемкость	Общая трудоемкость (акад. часов)	В том числе, аудиторных		Промежуточная аттестация	СРС
				Лекций	Практических занятий		
1	Модуль 1. Добро пожаловать в мир Swift	6	4	2	2	Лабораторная работа	2
2	Модуль 2. Основы	6	4	2	2	Лабораторная работа	2
3	Модуль 3. Базовые операторы. Строки	6	4	2	2	Лабораторная работа	2
4	Модуль 4. Коллекции. Выполнение ходом выполнения	6	4	2	2	Лабораторная работа	2
5	Модуль 5. Функции и замыкания	6	4	2	2	Лабораторная работа	2
6	Модуль 6. Выполнение итоговой работы	4	4		4	Лабораторная работа	

Итого:	36	24	10	14		12
Итоговая аттестация		Практическое задание				

Для всех видов аудиторных занятий академический час устанавливается продолжительностью 45 минут.

Форма Промежуточной аттестации – см. в ЛНА «Положение о проведении промежуточной аттестации слушателей и осуществлении текущего контроля их успеваемости» п.3.3

3. Календарный учебный график

Календарный учебный график формируется при осуществлении обучения в течение всего календарного года. По мере набора групп слушателей по программе составляется календарный график, учитывающий объемы лекций, практики, самоподготовки, выезды на объекты.

Неделя обучения	1	2	3	4	5	6	7	Итого часов
	пн	вт	ср	чт	пт	сб	вс	
1 неделя	8		8		8ИА	-	-	24
СРС	3	3	3	3				12
Итого:								36
Примечание: ИА – Итоговая аттестация (практическое задание)								

4. Рабочие программы учебных предметов

Модуль 1 . Добро пожаловать в мир Swift

- О языке Swift. Его история. Ресурсы.
- Роль и место Swift в стеке технологий Apple
- Первое приложение
- Инструменты
 - Playground
 - Xcode

Модуль 2 . Основы

- Константы и переменные
- Числовые типы
- Безопасность и определение типов
- Конвертация типов
- Логический тип
- Кортежи
- Опциональные значения

Модуль 3 . Базовые операторы. Строки

- Базовые операторы
- Операторы диапазона
- Логические операторы
- Строки и символы
- Мутабельность строк
- Работа с символами
- Конкатенация строк и символов
- Интерполяция строк
- Юникод

Модуль 4 . Коллекции. Управление ходом выполнения

- Мутабельность коллекций
- Массивы
- Словари
- Управление ходом выполнения
- Циклы
- Условные выражения

Модуль 5 . Функции и замыкания

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

Модуль 6 . Выполнение итоговой работы

- Тест
- Практическое задание

5. Организационно-педагогические условия

Соблюдение требований к кадровым условиям реализации дополнительной профессиональной программы:

а) преподавательский состав образовательной организации, обеспечивающий образовательный процесс, обладает высшим образованием и стажем преподавания по изучаемой тематике не менее 1 года и (или) практической работы в областях знаний, предусмотренных модулями программы, не менее 3 (трех) лет;

б) образовательной организацией наряду с традиционными лекционно-семинарскими занятиями применяются современные эффективные методики преподавания с

применением интерактивных форм обучения, аудиовизуальных средств, информационно-телекоммуникационных ресурсов и наглядных учебных пособий.

Соблюдение требований к материально-техническому и учебно-методическому обеспечению дополнительной профессиональной программы:

а) образовательная организация располагает необходимой материально-технической базой, включая современные аудитории, библиотеку, аудиовизуальные средства обучения, мультимедийную аппаратуру, оргтехнику, копировальные аппараты. Материальная база соответствует санитарным и техническим нормам и правилам и обеспечивает проведение всех видов практической и дисциплинарной подготовки слушателей, предусмотренных учебным планом реализуемой дополнительной профессиональной программы.

б) в случае применения электронного обучения, дистанционных образовательных технологий каждый обучающийся в течение всего периода обучения обеспечивается индивидуальным неограниченным доступом к электронной информационно-образовательной среде, содержащей все электронные образовательные ресурсы, перечисленные в модулях дополнительной профессиональной программы.

6. Формы аттестации и оценочные материалы

Образовательная организация несет ответственность за качество подготовки слушателей и реализацию дополнительной профессиональной программы в полном объеме в соответствии с учебным планом.

Оценка качества освоения дополнительной профессиональной программы слушателей включает текущий контроль успеваемости и итоговую аттестацию.

Результаты итоговой аттестации слушателей ДПП в соответствии с формой итоговой аттестации, установленной учебным планом, выставляются по двух бальной шкале («зачтено\незачтено»).

Слушателям, успешно освоившим дополнительную профессиональную программу и прошедшим итоговую аттестацию, выдается удостоверение о повышении квалификации.

Слушателям, не прошедшим итоговой аттестации или получившим на итоговой аттестации неудовлетворительные результаты, а также лицам, освоившим часть дополнительной профессиональной программы и (или) отчисленным из образовательной организации, выдается справка об обучении или о периоде обучения по образцу, самостоятельно устанавливаемому образовательной организацией.

Промежуточная аттестация проводится в форме контроля лабораторных и домашних заданий по завершении модуля .

Итоговая аттестация проводится по форме зачета практических и домашних заданий в соответствии с учебным планом. Результаты итоговой аттестации заносятся в соответствующие документы.

Условия прохождения

Время(мин): 40

Равномерно распределение по модулям

Вопросы для Итоговой аттестации:

Грамматика курса обеспечена фондом оценочных средств для проведения входного, текущего контроля и промежуточной аттестации. Фонд включает задания к лабораторным работам, контрольным работам, вопросы к итоговой аттестации.

Используемые формы текущего контроля: аудиторские самостоятельные работы: типовые контрольные задания; лабораторные работы; устный опрос.

Вопросы к итоговой аттестации:

1. Алгоритмы: определение, свойства.
2. Жизненный цикл программного обеспечения.
3. Обзор языка программирования Swift.
4. Способы описания алгоритмов.
5. Структура программы на языке Swift.
6. Типы данных в языке Swift. Определение простых переменных в программе.
7. Классы памяти объектов языка Swift.
8. Арифметические операции языка Swift.
9. Логические операции языка Swift
10. Операции языка Swift: приведения типа, условная, перечисления, взятия размера.
11. Операторы языка Swift: условный, оператор-переключатель.
12. Операторы циклов в языке Swift.
13. Одномерные массивы в Swift: определение, способы инициализации.
14. Типовые операции с одномерными массивами: поиск минимума, сумма элементов
15. Двумерные массивы в языке Swift: определение, способы инициализации.
16. Указатели: назначение, определение, инициализация.
17. Динамическое выделение памяти.
18. Функции ввода-вывода в Swift
19. Универсальный метод выделения динамической памяти под двумерные массивы.
20. Строки в языке Swift: определение, инициализация, кодирование символов.
21. Функции для работы со строками языка Swift.
22. Структурный тип данных: определение типа и переменных, инициализация структур.
23. Вложенные структуры, массивы структур. Динамическое выделение памяти под массивы структур.
24. Битовые поля структур: определения, преимущества, ограничения при работе.
25. Перечислимый тип данных: определение, назначение, пример использования.
26. Динамические списки: назначение, основные принципы организации.
27. Односвязный список: определение, особенности использования, типовые алгоритмы обработки.
28. Сравнительный анализ односвязного списка и массива структур.
29. Двусвязный список: определение, особенности использования, типовые алгоритмы обработки.
30. Поточковый ввод-вывод. Определение потока, стандартные потоки, типы потоков.
31. Открытие потоков: режимы открытия, различия текстовых и двоичных потоков.
32. Основные функция контроля состояния потоков.
33. Основные функции ввода строковой информации в поток и вывода информации из потока.
34. Форматированный потоковый ввод-вывод в языке Swift.
35. Ввод-вывод с использованием буфера. Отличия от форматированного ввода-вывода.
36. Функции в Swift: определение, прототип, вызов.
37. Основные преимущества структурного стиля программирования.
38. Формальные и фактические параметры. Связь формальных и фактических параметров функций.
39. Порядок вызова функции и возврата из нее. Передача параметров по значению и по адресу.
40. Функции и массивы: массивы как параметры функций.
41. Функции с умалчиваемыми значениями параметров.
42. Передача параметров функции main.

43. Ссылки: назначение, определение, использование.
44. Рекурсивные вызовы функций: назначение, примеры рекурсивных задач, прямая и косвенная рекурсия.
45. Разворачивание и свертывание рекурсии в памяти, глубина рекурсии (на практическом примере).
46. Алгоритм быстрой обменной сортировки, его рекурсивная реализация
47. Указатели на функции: определение, назначение, примеры использования.
48. Указатели на функции как параметры функций, сортировка массивов с использованием `qsort`.
49. Перегрузка функций: определение, достоинства, ограничения. Примеры реализации.
50. Препроцессор: назначение, основные этапы препроцессорной обработки.
51. Препроцессор: директивы включения файлов и замены идентификаторов.
52. Препроцессор: макроопределения с параметрами.
53. Препроцессор: сравнение функций и макросов.
54. Препроцессор: директивы условной компиляции.
55. Основы объектно-ориентированной парадигмы программирования: инкапсуляция, наследование, полиморфизм.
56. Определение классов и объектов в программах на языке Swift
57. Область видимости компонент класса.
58. Правила наследования классов в языке Swift.

Примерные темы лабораторных работ:

Лабораторная работа № 1.

Тема: Разработка алгоритмов решения линейных и разветвляющихся вычислительных процессов.

Лабораторная работа №2.

Тема: Разработка алгоритмов решения циклических вычислительных процессов

Лабораторная работа № 3.

Тема: Знакомство с системой программирования на языке Swift

Лабораторная работа №4.

Тема: Линейная программа. Простые типы данных. Программирование разветвляющихся процессов.

Лабораторная работа №5. Тема: Программирование циклических процессов. Типовые алгоритмы.

Лабораторная работа №6.

Тема: Программирование алгоритмов обработки одномерных массивов.

Лабораторная работа №7.

Тема: Программирование алгоритмов обработки двумерных массивов.

Лабораторная работа №8.

Тема: Обработка строковых данных в программах на языке Swift.

Лабораторная работа № 9.

Тема: Использование структурного типа данных в программах на языке Swift.

Лабораторная работа № 10.

Тема: Программирование с использованием динамических списков данных.

Лабораторная работа № 11.

Тема: Сохранение данных в файлах на языке Swift.

Лабораторная работа № 12.

Тема: Работа с текстовыми файлами в языке Swift.

Лабораторная работа № 13.

Тема: Программирование с использованием функций, определяемых пользователем.

Лабораторная работа № 14.

Тема: Программирование рекурсивных алгоритмов на языке Swift.

Лабораторная работа № 15.

Тема: Разработка многомодульных приложений на языке Swift.

Примеры задач:

1. Перечислимый тип данных: определение, назначение, пример использования.

2. Ссылки: назначение, определение, использование.

3. Задача

Написать программу, которая при запуске с параметром `‘/a ‘` выдавала бы информацию о разработчике программы, с параметром `“/f=filename”` выводила бы файл с именем `filename` на экран, а при любом другом параметре, а также при запуске без параметра выдавала бы сообщение о неверном параметре запуска.

4. Вариативные структуры: обобщенное определение, преимущества, примеры использования.

5. Задача

Написать программу меняющую в массиве $A[n][n]$ местами столбцы, содержащие наибольший и наименьший элементы. Поиск максимума и минимума и перестановку столбцов оформить в виде функций.

6. Открытие потоков: режимы открытия, различия текстовых и двоичных потоков.

7. Указатели на функции: определение, назначение, примеры использования.

8. Задача

Дан массив $A(n,2)$ координат точек на плоскости. Найти наиболее отдаленные друг от друга точки (Вычисление расстояния между точками оформить в виде функции). Определить также функцию определения точки пересечения двух отрезков по координатам их концов. Используя эту функцию, определить, является ли данный многоугольник самопересекающимся

9. Основные функции ввода строковой информации в поток и вывода информации из потока.

10. Препроцессор: макроопределения с параметрами.

11. Порядок вызова функции и возврата из нее. Передача параметров по значению и по адресу.

12. Задача

Три прямые на плоскости заданы уравнениями $akx+bky=ck$, $k=1,2,3$. Если эти прямые попарно пересекаются и образуют треугольник, тогда найти его площадь. Вычисление площади и анализ прямых на пересечение оформить в виде функций.

13. Поточковый ввод-вывод. Определение потока, стандартные потоки, типы потоков.

14. Динамические списки: назначение, основные принципы организации.
15. Основные функция контроля состояния потоков.
16. Задача

Ввести и записать в файл несколько структур "Зачетка студента" с полями:

Фамилия студента

Номер зачетки

Массив структур "Предмет"

Структура "Предмет" имеет поля: Фамилия преподавателя, Название предмета, Оценка

Отсортировать студентов в файле по среднему баллу

17. Функции с умалчиваемыми значениями параметров.

18. Форматированный потоковый ввод-вывод в языке Swift.

19. Задача

Записать в файл несколько структур типа «зачетка студента» с полями:

Фамилия студента; Номер зачетки; Массив структур "Предмет"

Структура "Предмет" имеет поля: Фамилия преподавателя; Название предмета; Оценка и удалить из файла все зачетки троечников.

20. Перегрузка функций: определение, достоинства, ограничения. Примеры реализации.

21. Ввод-вывод с использованием буфера. Отличия от форматированного ввода-вывода.

22. Задача

Пусть функция $Socr(a,b,p,q)$ от целых параметров приводит дробь к несократимому виду
Описать данную функцию и использовать ее для приведения дроби $1+1/2+1/3 +...+1/n$ к несократимому виду .

23. Рекурсивные вызовы функций: назначение, примеры рекурсивных задач, прямая и косвенная рекурсия.

24. Препроцессор: сравнение функций и макросов.

25. Задача

В текстовом файле, имя которого вводится с клавиатуры, переформатировать содержимое таким образом, чтобы каждая строка файла была на 1 символ длиннее предыдущей.

26. Основные преимущества структурного стиля программирования.

27. Задача

В текстовом файле, имя которого вводится с клавиатуры, удалить все предложения, содержащие больше 5 слов.

28. Односвязный список: определение, особенности использования, типовые алгоритмы обработки.

29. Рекурсивные вызовы функций: назначение, примеры рекурсивных задач, прямая и косвенная рекурсия.

30. Задача

Дан массив отрезков $M[5]$. Для каждой тройки этих отрезков, из которых можно построить треугольник, напечатать площадь данного треугольника (определить функцию, печатающую площадь треугольника со сторонами x , y и z , если такой треугольник существует).

31. Двусвязный список: определение, особенности использования, типовые алгоритмы обработки.

32. Формальные и фактические параметры. Связь формальных и фактических параметров функций.

33. Задача

Отсортировать строки файла, имя которого вводится с клавиатуры, в порядке возрастания их длины.

34. Сравнительный анализ односвязного списка и массива структур.

35. Функции и массивы: массивы как параметры функций.

36. Задача

В текстовом файле, название которого вводится с клавиатуры, удалить во всех предложениях каждое третье слово.

37. Разворачивание и свертывание рекурсии в памяти, глубина рекурсии (на практическом примере).

38. Открытие потоков: режимы открытия, различия текстовых и двоичных потоков.

39. Задача

Реализовать в программе односвязный список, информационная часть элементов которого содержит поля:

 Фамилия студента

 Номер группы

 Средний балл за экзамены

Удалить из списка всех неуспевающих студентов

40. Препроцессор: директивы включения файлов и замены идентификаторов.

41. Передача параметров функции main.

42. Задача

Реализовать в программе двусвязный список, информационная часть элементов которого содержит поля:

 Фамилия, имя, отчество

 День, месяц, год рождения

 Адрес

Удалить из списка всех близнецов по гороскопу

43. Поточковый ввод-вывод. Определение потока, стандартные потоки, типы потоков.

44. Функции в Swift: определение, прототип, вызов.

45. Задача

Реализовать в программе два односвязных списка. Удалить из первого списка все элементы, которые есть и в первом и во втором списке

46. Указатели на функции как параметры функций, сортировка массивов с использованием qsort.

47. Алгоритм быстрой обменной сортировки, его рекурсивная реализация

48. Задача

Написать функцию, которая изменяет регистр символов строки-параметра с прописного на заглавный. Используя эту функцию изменить регистр каждой четной строки файла, имя которого вводится с клавиатуры

49. Функции в Swift: определение, прототип, вызов.

50. Препроцессор: директивы условной компиляции.

51. Задача

Создать файл целых чисел, удалить из файла все элементы, меньшие среднего арифметического всех чисел файла.