

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**  
Учебно-методическое объединение по естественнонаучному образованию



**УТВЕРЖДАЮ**

Первый заместитель Министра  
образования Республики Беларусь

И.А. Старовойтова

« 02 » 2020 г.

Регистрационный № ТД- Г.639 /тип.

**ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ**

**Типовая учебная программа по учебной дисциплине  
для специальности**

**1- 31 03 07 Прикладная информатика (по направлениям)**

**для направления специальности**

**1- 31 03 07-01 Прикладная информатика**

**(программное обеспечение компьютерных систем)**

**СОГЛАСОВАНО**

Председатель Учебно-методического объединения по естественнонаучному образованию

О.А. Ивашкевич

« 29 » 2018 г.

**СОГЛАСОВАНО**

Директор ООО «Техартгруп»

А.В. Кульгачев

« 27 » 2020 г.

**СОГЛАСОВАНО**

Начальник Главного управления профессионального образования Министерства образования Республики Беларусь

С.А. Касперович

« 13 » 02 2020 г.

**СОГЛАСОВАНО**

Проректор по научно-методической работе Государственного учреждения образования «Республиканский институт высшей школы»

И.В. Титович

« 10 » 02 2020 г.

Эксперт-нормоконтролер

О.И. Велишовский

« 4 » февраля 2020 г.

**СОСТАВИТЕЛИ:**

**А.Н.Курбацкий**, заведующий кафедрой технологий программирования Белорусского государственного университета, доктор технических наук, профессор;

**М.И.Давидовская**, старший преподаватель кафедры технологий программирования Белорусского государственного университета

**РЕЦЕНЗЕНТЫ:**

Кафедра интеллектуальных информационных технологий Учреждения образования «Белорусский государственный университет информатики и радиоэлектроники»;

**И.А.Король**, заместитель директора республиканского унитарного предприятия «Национальный центр электронных услуг», кандидат физико-математических наук, доцент

**РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ В КАЧЕСТВЕ ТИПОВОЙ:**

Кафедрой технологий программирования Белорусского государственного университета

(протокол № 12 от 17 мая 2018 г.);

Научно-методическим советом Белорусского государственного университета (протокол № 6 от 16 июня 2018 г.);

Научно-методическим советом по прикладной математике и информатике учебно-методического объединения по естественнонаучному образованию (протокол № 16 от 29 мая 2018 г.).

Ответственный за редакцию: М.И.Давидовская

Ответственный за выпуск: М.И.Давидовская

## Пояснительная записка

Типовая учебная программа по учебной дисциплине «Технологии программирования» разработана для учреждений высшего образования Республики Беларусь в соответствии с требованиями образовательного стандарта высшего образования первой ступени по специальности 1-31 03 07 «Прикладная информатика».

Учебная дисциплина «Технологии программирования» знакомит студентов с современными технологиями разработки программных продуктов, включая системы контроля версий, базы данных на примере SQLite, гибкие методологии управления программными проектами, методы и технологии разработки приложений для платформы iOS. Особое внимание уделяется вопросам управления проектами, написанию запросов на языке SQL, разработке приложений для платформы iOS.

В учебной дисциплине также рассматривается содержание основных этапов процесса создания программного продукта для платформы iOS, в том числе: проектирование модели предметной области и интерфейса приложения, создание базы данных на основе спроектированной модели, разработка и тестирование приложения, используя инструментарий IDE Xcode.

В учебной дисциплине «Технологии программирования» используются учебные материалы образовательной инициативы корпорации Apple – iOS Developer University Program (<https://developer.apple.com/programs/ios/university/>), – и материалы образовательной программы проекта GitHub – GitHub Education (<https://education.github.com/>). Выполнение лабораторных работ осуществляется в виртуализированном окружении с операционной системой macOS.

Основой для изучения данной дисциплины являются учебные дисциплины «Программирование» государственного компонента учебного плана, «Системное программирование» компонента учреждения высшего образования. Методы, излагаемые в учебной дисциплине, используются при изучении ряда дисциплин специализации, а так же дисциплин «Программирование мобильных и встраиваемых систем», «Проектирование человеко-машинных интерфейсов». Изучение технологий программирования и разработки приложений для платформы iOS позволяет студентам получить знания, необходимые им в дальнейшем для успешной работы по специальности.

**Цели преподавания учебной дисциплины «Технологии программирования»:**

– формирование у студентов системы знаний по проектированию и разработке программных продуктов на основе использования технологий и решений для платформы iOS, необходимых им в дальнейшем для успешной работы;

– формирование представлений об основах проектирования и разработки приложений для платформы iOS, ознакомление с языком программирования Objective-C и изучение языка программирования Swift.

**Основные задачи, решаемые при изучении дисциплины «Технологии программирования»:**

– изучение основ проектирования программных продуктов;

- изучение основ проектирования баз данных;
- изучение технологий управления проектами;
- изучение языков программирования Objective-C и Swift;
- изучение технологий разработки программного обеспечения для платформы iOS.

В результате изучения дисциплины студент должен:

**знать:**

- методы проектирования программных систем;
- методы проектирования интерфейса пользователя;
- методы разработки программных продуктов с использованием различных инструментальных средств;
- методы проектирования баз данных;
- основные понятия и методы тестирования программного обеспечения;

**уметь:**

- анализировать требования, предъявляемые к программным системам, составлять техническое задание, моделировать программное обеспечение, а также проводить его тестирование;
- проводить оценку качества программного обеспечения;
- проектировать интерфейс пользователя приложений;
- разрабатывать программные комплексы с использованием различных инструментальных средств;
- управлять версиями приложений;

**владеть:**

- современными методами создания и анализа программных комплексов.

**Требования к компетенциям специалиста.**

Специалист должен:

Уметь применять базовые научно-теоретические знания для решения теоретических и практических задач.

Владеть системным и сравнительным анализом.

Владеть исследовательскими навыками.

Уметь работать самостоятельно.

Владеть междисциплинарным подходом при решении проблем.

Быть способным порождать новые идеи (обладать креативностью).

Иметь навыки, связанные с использованием технических устройств, управлением информацией и работой с компьютером.

Обладать способностью к межличностным коммуникациям.

Уметь работать в команде.

Специалист должен быть способен:

**Проектно-конструкторская деятельность**

Проектировать, разрабатывать и тестировать программное обеспечение различных видов.

Разрабатывать техническую документацию на программное обеспечение.

Проектировать, разрабатывать системы баз данных.

**Научно-исследовательская деятельность**

Применять профессиональные знания и навыки для проведения научных

исследований в области прикладной информатики.

Разрабатывать и совершенствовать методы исследований в области информационных и телекоммуникационных систем.

Работать с научно-технической информацией с использованием современных информационных технологий.

Пользоваться глобальными информационными ресурсами.

#### **Экспертно-консультационная деятельность**

Оказывать консультации по вопросам работы программного обеспечения, в том числе разработанного сторонними организациями.

Анализировать результаты работы установленного программного обеспечения и выработать предложения по улучшению качества его работы.

Проводить обучение специалистов, занимающихся эксплуатацией программного обеспечения.

#### **Организационно-управленческая деятельность**

Взаимодействовать со специалистами смежных профилей.

Готовить доклады, материалы к презентациям.

#### **Инновационная деятельность**

Осуществлять поиск, систематизацию и анализ информации по перспективам развития отрасли, инновационным технологиям, проектам и решениям.

Для изучения учебной дисциплины отведено 104 учебных часа, из них 68 – аудиторные занятия. Примерное распределение по видам занятий: лекции – 34 часа, лабораторные занятия – 34 часа.

### Примерный тематический план

| Название раздела, темы   | Количество аудиторных часов |             |                      |
|--|-----------------------------|-------------|----------------------|
|  | Всего                       | В том числе |                      |
|  |                             | Лекции      | Лабораторные занятия |
| <i>Раздел 1. Технологии разработки программного обеспечения и управление проектами</i>                   | <b>22</b>                   | <b>10</b>   | <b>12</b>            |
| <b>Тема 1.1.</b> Введение в разработку для платформы macOS и iOS   | 4                           | 2           | 2                    |
| <b>Тема 1.2.</b> Система контроля версий git и сборка приложений   | 8                           | 4           | 4                    |
| <b>Тема 1.3.</b> Разработка приложений и хранение данных в базе данных sqlite                            | 8                           | 2           | 6                    |
| <b>Тема 1.4.</b> Управление ИТ-проектами и гибкие методологии  | 2                           | 2           |                      |
| <i>Раздел 2. Разработка мобильных приложений под платформу iOS</i>                                       | <b>46</b>                   | <b>24</b>   | <b>22</b>            |
| <b>Тема 2.1.</b> Введение в разработку приложений для платформы iOS                                      | 2                           | 2           |                      |
| <b>Тема 2.2.</b> Основы языка программирования Objective-C   | 6                           | 2           | 4                    |
| <b>Тема 2.3.</b> Базовые возможности языка Swift   | 6                           | 4           | 2                    |
| <b>Тема 2.4.</b> Язык Swift, объектно-ориентированное программирование и функциональное программирование | 4                           | 2           | 2                    |
| <b>Тема 2.5.</b> Проектирование интерфейса приложений и локализация                                      | 4                           | 2           | 2                    |
| <b>Тема 2.6.</b> Графика, анимация и жесты   | 8                           | 4           | 4                    |
| <b>Тема 2.7.</b> Хранение данных в iOS   | 8                           | 4           | 4                    |
| <b>Тема 2.8.</b> Определение местоположения, другие возможности и библиотеки                             | 8                           | 4           | 4                    |
| <b>Всего</b>   | <b>68</b>                   | <b>34</b>   | <b>34</b>            |

## Содержание учебного материала

### Раздел 1. Технологии разработки программного обеспечения и управление проектами

#### Тема 1.1. Введение в разработку для платформы macOS и iOS

Обзор современных операционных систем. История развития macOS и iOS. Технологии виртуализации и запуск macOS в виртуальной машине.

Архитектура и экосистема macOS. Пакетный менеджер и способы установки приложений. Командный интерпретатор. Базовые команды, права доступа. Файловые системы HFS+ (Mac OS Extended), APFS и уровни доступа.

#### Тема 1.2. Система контроля версий git и сборка приложений

Введение в системы контроля версий (СКВ). Типы СКВ. Распределенная система контроля версий git. Установка git. Создание и инициализация репозитория. Клонирование репозитория. Запись изменений в репозиторий. Внешние репозитории. Публикация изменений. Управление ветками. Консольный git клиент. Графические клиентские приложения управления версиями.

Разработка приложений на языке C и сборка консольных приложений. Файл Makefile. Утилиты make, cmake. Современные решения для автоматической сборки и распространения. Технологии непрерывной интеграции.

#### Тема 1.3. Разработка приложений и хранение данных в базе данных sqlite

Введение в системы управления базами данных. Основы sqlite. Консоль sqlite и основные команды управления базой данных (БД). Проектирование БД и её наполнение. Выборки и параметрические запросы. Триггеры в sqlite.

Разработка приложений на языке C и хранение данных в БД sqlite. Создание базы данных, таблиц, вставка строк и удаление. Выборки данных.

#### Тема 1.4. Управление ИТ-проектами и гибкие методологии

Введение в управление ИТ-проектами. Гибкие методологии разработки программного обеспечения. Базовые термины проектного управления. Принципы и ценности Agile. Обзор популярных гибких методологий на примере Scrum, Lean, Kanban и др. Применение Kanban для управления проектами.

### Раздел 2. Разработка мобильных приложений под платформу iOS

#### Тема 2.1. Введение в разработку приложений для платформы iOS

Знакомство с платформой iOS и функционалом iPhone и iPad. Архитектура iOS и слои абстракции, включая CoreOS, Core Services, Media и Cocoa Touch. Основные классы библиотеки CocoaTouch и жизненный цикл приложения. Концепция MVC в iOS-разработке.

Инструменты для разработки iOS-приложений. Возможности IDE Xcode. Конфигурирование и настройка проекта, файл списка свойств формата .plist. Использование симулятора. Редактирование и навигация по коду. Документация и комментирование кода. Управление версиями, подключение локальных и внешних репозиториями. Отладка в Xcode. Правила публикации приложений в AppStore.

### **Тема 2.2. Основы языка программирования Objective-C**

Объекты и синтаксис отправки сообщений. Переменные и объявление переменных. Атрибуты свойств и ограничение доступа к переменной. Строки и управление изменяемыми и неизменяемыми строками. Коллекции в Objective-C. Классы, объявление и реализация классов. Протоколы. Управление памятью.

### **Тема 2.3. Базовые возможности языка Swift**

Знакомство с основами языка и основными типами данных. Переменные, константы и коллекции данных. Операторы. Управляющие конструкции. Циклы. Ветвления. Функции. Комплексные типы данных: структуры; перечисления; свойства и методы; конструктор.

### **Тема 2.4. Язык Swift, объектно-ориентированное программирование и функциональное программирование**

Классы и объекты. Наследование. ARC и введение в управление памятью. Требования к классам, протоколы, расширения, полиморфизм, композиция. Замыкания. Функции высшего порядка.

### **Тема 2.5. Проектирование интерфейса приложений и локализация**

Основные элементы управления и их виды. Представления View и их виды. Привязка элементов интерфейса и обработка событий. ViewController и его жизненный цикл. Адаптация элементов интерфейса под разные устройства с помощью AutoLayout. Использование ресурсов приложения (изображения, прочие данные). Понятие делегирования. Обработка ввода текста. Проектирование макетов и навигации на примере UITableView, UICollectionView и других представлений. Навигация по приложению (TabBarController и UINavigationController).

Интернационализация и локализация. Локализация интерфейса и файлы локализации. Локализация макетов storyboard. Интернационализация изображений и форматов данных. Словарь конфигурации спецификатора спецификации формата (поддержка множественного числа). Добавление нового языка. Локализация иконки приложения.

### **Тема 2.6. Графика, анимация и жесты**

Библиотека Core Graphics и ее основные возможности. Геометрические типы данных. Графические примитивы и двумерная графика. Цвета и шрифты. Рисование. Использование камеры и фотоархива. Селектор изображений и

Анимация представлений и анимация слоев. Возможности библиотеки Core Animation. Анимация и анимируемые свойства UIView. Анимация с помощью CALayer и его наследников.



Жесты и виды жестов. Обработка жестов. Реагирующий элемент и цепочка реагирующих элементов. Мультисенсорная архитектура. Методы уведомления о касаниях. Обработка одного и нескольких жестов. Распознавание жестов и `UIGestureRecognizer`.

### **Тема 2.7. Хранение данных в iOS**

Механизмы хранения данных в файловой системе iOS: списки свойств, архивы объектов, `SQLite3`, `Core Data`. Каталоги приложения (`Documents`, `Library` и `tmp`), определение местоположения и права доступа. Стратегии хранения файлов их их преимущества и недостатки.

Списки свойств. Запись и считывание. Сериализация списка свойств. Управление настройками приложения и класс `UserDefaults` (`NSUserDefaults`).

Архивирование объектов моделей. Протоколы `NSCoding` и `NSCopying`. Шифрование и восстановление заархивированного объекта на уровне класса и суперкласса.

`SQLite3` и способы подключения библиотеки в проект. Открытие и закрытие базы данных. Создание и наполнение таблиц БД. Извлечение данных из БД. Связанные переменные.

Библиотека `Core Data`. Сущности и управляемые объекты. Свойства сущностей: атрибуты, связи и извлекаемые свойства (`fetchd properties`). Извлечение значений из атрибута и запись в атрибут. Контекст и использование контекста. Постоянное (`persistence store`) и резервное (`backing data store`) хранилища. Контекст управляемых объектов и менеджер отмены. Создание и извлечение управляемых объектов. Разработка модели данных.

### **Тема 2.8. Определение местоположения, другие возможности и библиотеки**

Библиотека `Core Location`. Определение местоположения, основные подходы и методы. Разрешение на использование служб определения местоположения. Диспетчер местоположения `CLLocationManager`. Запуск и использование диспетчера местоположения. Требуемая точность определения местоположения и фильтры расстояний. Обновление координат и определение расстояния между объектами `CLLocation`.

Библиотека `MapKit`. Визуализация карты. Понятие маркера и `MKAnnotation`. `MKDirections` (направления) и `MKOverlay` («наложения»). Методы делегата `MKMapViewDelegate`. Автоматическая группировка маркеров при масштабировании.

Распознавание ориентации и перемещения устройства. Работа с сетью. Документы и служба `iCloud`. Отправка сообщений (`email`, `SMS`). Интеграция с социальными сервисами. Подключение сторонних библиотек.

Модульное тестирование. Преимущества и недостатки модульного тестирования. Разработка модульных тестов. Непрерывная интеграция приложений для платформы iOS.

## Информационно-методическая часть

### Литература

#### *Основная*

1. Кочан, Ст. Программирование на Objective-C / Ст. Кочан. – изд. 6, доп. и перераб. – М.: ЭКОМ Паблишерз, 2014. – 550 с.
2. Марк, Д. Swift. Разработка приложений в среде Xcode для iPhone и iPad с использованием iOS SDK / Д. Марк, Дж. Наттинг [и др] – М.: Вильямс, 2016. – 816 с.
3. Маскри, М. Swift 3. Разработка приложений в среде Xcode для iPhone и iPad с использованием iOS SDK / М. Маскри, К. Топли – М.: Вильямс, 2017. – 896 с.
4. Нойбург, М. Программирование для iOS 7. Основы Objective-C, Xcode и Cocoa / М. Нойбург. – М.: Вильямс, 2014. – 384 с.
5. Усов, В. Swift. Основы разработки приложений под iOS и MacOS / В. Усов. – СПб.: Питер, 2017. – 368 с.
6. Чакон, С. Git для профессионального программиста / С. Чакон, Б. Штрауб. – СПб.: Питер, 2017. – 496 с.

#### *Дополнительная*

1. Грей, Эн. Swift. Карманный справочник. Программирование в среде iOS и OS X / Эн Грей. – М.: Вильямс, 2016. – 288 с.
2. Гэлловей, М. Сила Objective-C 2.0. Эффективное программирование для iOS и OS X / М. Гэлловей. – СПб.: Питер, 2014. – 304 с.

## **Перечень рекомендуемых средств диагностики**

Текущий контроль по дисциплине «Технологии программирования» рекомендуется осуществлять в течение процесса обучения в виде проверки лабораторных работ, вопросов для самоконтроля, автоматизированных тестов и проведения коллоквиумов (лекционная часть курса).

Текущий контроль самостоятельной работы студентов по усвоению знаний по учебной дисциплине «Технологии программирования» рекомендуется осуществлять преподавателем на аудиторных занятиях в виде выполнения тестовых заданий.

Рекомендуемая форма текущей аттестации – автоматизированное тестирование.

Рекомендуемая форма итогового контроля – экзамен.

При этом рекомендуется оценивать успеваемость на основе модульно-рейтинговой системы.

## **Методические рекомендации по организации и выполнению самостоятельной работы**

Условия для самостоятельной работы студентов, в частности, для развития навыков самоконтроля, способствующих интенсификации учебного процесса, обеспечиваются наличием и полной доступностью электронных (и бумажных) учебно-методических пособий по основным разделам учебной дисциплины.