

Программа обучения для школьников

**на 2019-2020 учебный год**

Москва 2020

[Программа «IT ШКОЛА SAMSUNG»](http://www.samsung.com/ru/itschool/) компании Samsung Electronics - это долгосрочная

инициатива, которая реализуется при поддержке Министерства Образования и Науки РФ. Программа работает с 2014 года. Более 2800 школьников получили сертификат Samsung, успешно завершив программу дополнительного образования в области IT и

программирования за 5 лет. В 2019 году Samsung значительно расширил географию проекта – программа IT ШКОЛА SAMSUNG будет работать в более 40 регионах России.

Разработчики программы - Samsung R&D Institute Rus (Исследовательский Центр Samsung) при участии Московского физико-технического института (МФТИ).

Программа ориентирована на школьника 9-10 класса, который:

* имеет склонность к алгоритмическому мышлению, увлекается ИТ-технологиями;
* владеет хотя бы одним языком программирования на уровне знания условных и циклических конструкций;
* имеет устойчивые знания по школьному курсу математики.

Особенности программы:

* Комплексный подход в построении курса: охвачены минимально необходимые базовые знания из разных областей ИТ и программирования с учетом уровня подготовки

школьников.

* Проектное обучение: разработка учащимся индивидуального проекта в виде приложения для мобильных платформ на ОС Android.
* Ориентация на группы/учеников с различным уровнем подготовки: выделение в

программе основного «ядра» и тем/материалов для дополнительного углубленного изучения (\*).

Продолжительность обучения 1 учебный год, 132 ак. часа, из них: 122 ч. - аудиторные занятия по 4 часа в неделю, 2 часа – защита проекта, 8 часов – консультации по выполнению

индивидуальных проектов (в программе рекомендуемые темы консультаций помечены \*\*).

Программа обучения состоит из пяти модулей:

1. Основы программирования на языке Java - 20 часов
2. Введение в объектно-ориентированное программирование - 24 часа
3. Основы программирования Android приложений - 24 часа
4. Алгоритмы и структуры данных на языке Java - 32 часов
5. Основы разработки серверной части мобильных приложений - 22 часа

Виды деятельности учащегося в процессе обучения:

* освоение теоретического и практического материала на занятиях
* самостоятельная практическая работа: выполнение домашних заданий, мини-проектов (небольшие приложения, которые реализуются учениками преимущественно на занятиях совместно с учителем с небольшими самостоятельными доработками в качестве домашнего задания)
* разработка индивидуального проекта
* участие в вебинарах
* промежуточная помодульная аттестация в форме электронного тестирования Далее приведено тематическое содержание модулей.

|  |  |  |
| --- | --- | --- |
| **Ча- сов** | **№** | **Описание темы** |
| **20** | **1** | **Модуль 1. Основы программирования на языке Java** |
| 2 | 1.1 | **Здравствуй мир!**  Знакомство с системой обучения в IT ШКОЛЕ SAMSUNG.  Порядок создания, компиляции, сборки и запуска в IDE приложения Java. Порядок инсталляции IDE в домашних условиях.  Выбор IDE по усмотрению преподавателя: либо IntelliJ IDEA + Android Studio или Eclipse + ADT. Последний вариант не рекомендуется использовать при работе в ОС Windows |
| 2 | 1.2 | **Типы данных и операции**: переменные, примитивные типы данных, арифметические операторы, выражения и присваивания |
|  | 1.3\*1 | **Представление данных в памяти**: представление числовых (в прямом коде) и символьных данных |
|  | 1.4\* | **Представление отрицательных и вещественных чисел в памяти.** Представление чисел в обратном и дополнительном коде  **Поразрядные операции** |
| 2 | 1.5 | **Логические выражения**. Тип Boolean. Операции отношения, логические и тернарные операции |
| 2 | 1.6 | **Условные конструкции.** Понятие блока, область действия блоков. Условные конструкции: if-else, switch |
| 2 | 1.7 | **Итеративные конструкции while, do-while**. Безусловные операторы перехода break |
| 2 | 1.8 | **Итеративные конструкции for**. **Массивы**.  Вложенные циклы. Безусловные операторы перехода break с меткой, continue.  Одномерные массивы, цикл for each в Java. Разбор примеров нахождения максимума и минимума, поиска на числовых массивах |
| 2 | 1.9 | **Методы**. Изучение понятия функций на примере методов Java. Передача параметров, возвращение результата. Видимость переменных |
|  | 1.10\* | **Указатели на примере языка Си**: понятие, использование в качестве аргументов  функций. Методы передачи аргументов в функцию. Динамическое выделение памяти |
| 2 | 1.11 | **Многомерные массивы**. Неровные массивы\* |
| *2* | *1.12* | *Практикум* |
| *2* | *1.13* | *Контрольное тестирование по модулю* |

1 Обозначение \*- дополнительные темы, которые могут быть включены в основную программу обучения по

усмотрению преподавателя

|  |  |  |  |
| --- | --- | --- | --- |
| **24** | **2** | **Модуль 2. Введение в объектно-ориентированное программирование** | |
|  |  | **Android практикум** |
| 2 | 2.1 | **Понятие класса и объекта**.  Цели и задачи ОО-подхода к проектированию и разработке ПО. Объект, сообщение, класс, экземпляр объекта, метод.  Общее понятие о парадигмах ООП: абстракция, инкапсуляция, наследование, полиморфизм. Описание протокола класса.  Обзор классов, соответствующих примитивным типам |  |
| 4 | 2.2 | **Работа с экземплярами класса**  Конструкторы и деструкторы. Статические методы. Открытые и закрытые поля. Доступ к полям объекта. Инициализация переменных, массивов, полей классов в конструкторе. Конструктор по умолчанию. Перегрузка  методов на примере конструкторов |  |
| 2 | 2.3 | **Строки. Основы тестирования и отладки**  Начальные приемы тестирования и отладки, сценарии тестирования на примерах со строками |  |
| 2 | 2.4 | **Знакомство с Android разработкой**  ОС Android. Среда разработки. Принципиальная архитектура Android-приложения | Создание первого  Android приложения в IDE. Изучение  жизненного цикла Activity |
| 4 | 2.5 | **Интерфейс Android приложения**  Построение простейшего интерфейса пользователя. Язык разметки XML. Описание ресурсов Android с помощью XML.  Разметки (Layouts) и их применение. Представления (Views) | Создание приложения с простейшим  интерфейсом и  обработкой событий |
| 4 | 2.6 | **Наследование, инкапсуляция и полиморфизм**  Производные классы и наследование. Защищенные части классов и правила доступа для классов и объектов в Java. Сравнение иерархии классов и контейнеризации классов. Полиморфные методы и позднее связывание в Java.  Абстрактные методы и классы, интерфейсы. | Создание класса, хранящего  информацию о  человеке (возраст, имя).  Реализация иерархии классов  «Геометрические фигуры» |
| 2 | 2.7 | **Context и Intent**  Понятия контекста (Context) и намерения (Intent) в Android | Намерения (Intents) в Android |
|  | 2.8\* | **Параметризованные типы** (Generic) как пример статического полиморфизма в Java | Использование Generic |
| 2 | 2.9 | Практикум | |
|  | *2.10\**  *\* 2* | *Формирование постановки задачи на индивидуальный проект (срок публикации в системе обучения темы индивидуального проекта – окончание 2 модуля)* | |
| *2* | *2.11* | *Контрольное тестирование по модулю* | |

2 Обозначение \*\* - занятия для организации работы учащихся по выполнению индивидуальных проектов

(проводятся по усмотрению преподавателя)

|  |  |  |  |
| --- | --- | --- | --- |
| **24** | **3** | **Модуль 3. Основы программирования Android приложений** | |
|  |  | **Android практикум** |
| 4 | 3.1 | **Практикум ООП проектирования**.  Разбор кейсов проектирования архитектуры классов приложения. Диаграммы UML | Разбор задания минипроекта  3.1. Получение диаграммы классов мини-проекта |
| 2 | 3.2 | **Ввод-вывод в Java. Исключения**  Библиотечные классы ввода-вывода. Обработка исключений и классы исключений. Стандартные  исключения Java. Работа с файлами в Android | Работа с классом File, как пример необходимости  обработки исключений. |
| 4 | 3.3 | **Внутренние и анонимные классы** на примерах обработчиков событий пользовательского  интерфейса | Разбор примеров  использования Listener  Работа над мини-проектом 3.1 |
| 2 | 3.4 | **Параллелизм и синхронизация**.  Процессы и потоки в Android. Классы AsyncTask и Thread. Реализация логики потоков. Синхронизация  потоков | Разбор примера  использования AsyncTask Работа над мини-проектом  3.1 |
|  | 3.5\* | **Сервисы в Android**.  Жизненный цикл сервисов и управление им. Класс IntentService | Разбор примеров |
| 2 | 3.6 | **Фрагменты** (Fragments). **Сенсоры**  Создание и управление фрагментами. Класс  Fragment и его методы. Взаимодействие фрагментов и активностей. Типы сенсоров и обработка событий | Разбор примера изменения интерфейса с фрагментами с учетом данных гироскопа |
| 2 | 3.7 | **Двумерная графика** в Android приложениях. Класс Canvas | Реализация простейшего приложения на Canvas |
| 4 | 3.8 | **Разработка игровых приложений**.  Этапы проектирования и реализации. Профессии в мире индустрии игр. Понятие игрового движка.  Реализация графики на основе SurfaceView | Разбор игры с анимацией на SurfaceView |
|  | 3.9\* | **Разработка 3D игр с использованием фреймворка libGDX**  <https://habrahabr.ru/post/276139/> | Простейший пример с  анимацией, HUD, обработкой касаний |
| 2 | 3.10 | Практикум |  |
|  | *3.11\*\** | *Защита прототипа индивидуального проекта* | |
| *2* | *3.12* | *Контрольное тестирование* | |

|  |  |  |  |
| --- | --- | --- | --- |
| **32** | **4** | **Модуль 4. Алгоритмы и структуры данных** | |
|  |  | **Практикум** |
| 4 | 4.1 | **Массив**, как базовая структура данных. Алгоритм двоичного поиска: идея, применения, реализация.  Классы Arrays (массивы), ArrayList (неограниченный массив) | Изучение класса Arrays:  заполнение, копирование,  сравнение, печать, методы поиска и др. общие методы.  Изучение класса ArrayList.  Создание итераторов, навигация |
| 4 | 4.2 | **Список**, как базовая структура данных: стеки, очереди, односвязные и двусвязные списки | Изучение класса LinkedList, реализующему связные списки |
| 2 | 4.3 | **Адаптеры в Андроид**  Назначение и применение. Стандартные адаптеры | Использование готовых адаптеров ArrayAdapter и  SimpleAdapter для реализации ListView |
| 2 | 4.4 | **Реляционная модель данных** и реляционные схемы. Представление данных в виде таблиц. Типы связей: один к одному, один ко многим,  многие-ко-многим. Проектирование простейшей БД  Необходимость возникновения и история развития СУБД. Обзор и классификация современных СУБД | Разбор задания на минипроект по БД 4.1 (Чемпионат по футболу)  Проектирование схемы БД минипроекта |
| 4 | 4.5 | **Локальная СУБД** на примере SQLite.  Введение в SQL. Создание и наполнение таблиц: команды CREATE, INSERT.  Команда SELECT для выборки данных, UPDATE для изменения, DELETE для удаления записей из таблицы. Дополнения к запросам SELECT.  Ключевые слова ORDER BY, DISTINCT. Агрегация в SELECT в запросах: COUNT, SUM, AVG, MAX, MIN | Знакомство с SQLite. Разбор  заготовки Android приложения. Реализация изученных команд SQL в минипроекте 4.1 |
| 2 | 4.6 | **Рекурсия**  Линейная и ветвящаяся рекурсия. Стек вызовов | Пример использования при обходе дерева директорий |
| 2 | 4.7 | **Дерево**, как базовая структура данных.  Сбалансированные деревья. Двоичные деревья | Практическое занятие по библиотечному классу TreeSet |
| 2 | 4.8 | **Обзор алгоритмов сортировок**: пузырьковая, вставкой и быстрая в сравнении трудоемкости. **Компараторы** | Компаратор, интерфейс Comparable. Практическое  занятие по использованию методов класса Arrays, реализующих сортировку |

|  |  |  |  |
| --- | --- | --- | --- |
| 2 | 4.9 | **Хэш-таблица и функция хэширования**  Поддержка хэширования в Java, метод hashCode. Семейства контейнеров Collections и Map | Практическое занятие по  библиотечным классам HashSet и TreeSet |
| 2 | 4.10 | **Ассоциативные массивы**.  Класс Map, контейнеры HashMap, TreeMap. Хранение данных в Android Preferences | Практическое занятие по библиотечным классам,  реализующим ассоциативные контейнеры |
|  | 4.11\* | **Контент-провайдеры** в Android.  Назначение и применение. Стандартные  контент-провайдеры ContactsContract, MediaStore | Практика использования стандартных контент-  провайдеров на примере контактов и медиа-файлов |
|  | 4.12\* | **Введение в криптографию и криптоанализ**. Простейшие алгоритмы подстановочного шифрования | Практическое занятие на распознавание сообщения, зашифрованного  подстановочным шифром |
|  | 4.13\* | **Введение в шифрование**  Понятие ключа шифрования. Симметричные и  асимметричные ключи. Алгоритм DES. Алгоритм RSA. Примеры использования. | Практическое занятие на  освоение ключей шифрования. |
| 4 | 4.14 | Практикум | |
|  | 4.15\*\* | *Предзащита индивидуального проекта* | |
| *2* | 4.16 | *Контрольное тестирование* | |

|  |  |  |  |
| --- | --- | --- | --- |
| **22** | **5** | **Модуль 5. Основы разработки серверной части мобильных приложений** | |
|  |  | **Практикум** |
| 2 | 5.1 | **IP-сети**  Адресация в IP-сетях. IPv4. Автоматизация назначения IP-адресов (DHCP). Доменные имена (DNS), URL- ссылки. Несколько IP адресов для одного сайта.  Популярные сетевые команды ping, tracert, ipconfig. Сервисы работы с IP-адресами. | Работа с сетевыми командами и сервисами |
| 4 | 5.2 | **Web сервер. HTTP запросы и ответы**  Протокол HTTP. Понятие web-сервера. Запросы клиента POST и GET, коды ответов сервера, заголовки запросов и ответов.  Разбор запросов и ответов сервера. | Отправка запроса на север с  помощью формы из браузера. Обработка методов на сервере средствами Java,  \*PHP. |
| 4 | 5.3 | **Клиент-серверная архитектура мобильных**  **приложений:** структура, схема взаимодействия сервера и клиента мобильного приложения. Формат JSON и XML. Сериализация. Библиотека Retrofit. | Отправка запросов из Android приложения. Реализация сервера средствами Java,  \*PHP. |
| 4 | 5.4 | **Облачные платформы. REST взаимодействие** Облачные сервисы3 для хостинга серверной части  приложений. Реализация серверной части средствами  Java, \*PHP. Стиль взаимодействия REST. | Реализация Android приложения REST |
|  | 5.5\* | **Серверные СУБД**  Хранение данных на сервере в виде БД. Индексы. Знакомство с серверной СУБД на примере  PostgreSQL/MySQL. Реализация back-end части web сервера средствами Java, PHP. | Реализация БД в СУБД PostgreSQL на  <https://www.heroku.com/>и Android клиента к ней. |
|  | 5.6\* | Дизайн программного обеспечения и приложений  **Material Design** | Разбор примеров |
| 2 | 5.7 | *Контрольное тестирование* |  |
| 6 | 5.8 | Практикум. Работа над индивидуальным проектом | |
| **2** |  | **Защита проектов** |  |

3 В случае невозможности использования облачного сервиса из-за неустойчивого интернет-соединения в классе рекомендуем установить виртуальную машину со всем необходимым ПО. Образ доступен в курсе для

преподавателей.