

Министерство образования Красноярского края
Филиал АНО ДТ «Красноярский «Кванториум» в г. Норильске
«Центр цифрового образования детей IT-Куб г. Норильск»

УТВЕРЖДАЮ

Директор

Е.О. Герасименко

Приказ № 02-02-03/2

от «25» сентября 2020 г



Дополнительная общеобразовательная общеразвивающая
программа технической направленности

«Мобильная разработка»

Срок реализации:

1 семестр

Возраст детей:

13-17 лет

Составитель программы:

В.М. Поляков

Норильск, 2020 г.

Программу составил(и):

педагог дополнительного образования

Владислав Максимович Поляков

Рецензент(ы):

Заведующий кафедрой «Информационные
системы и технологии»,
ФГБОУ ВО «НГИИ»,
к.э.н., доцент



Михаил Вадимович Петухов

**ДООД одобрена на заседании методического совета
Филиал АНО «Красноярский «Кванториум» в г. Норильске
«Центр цифрового образования детей ИТ-Куб г. Норильск»**

Протокол от 30.09.2020 №1

Председатель методического совета
заведующий по учебной части

Елена Вячеславовна Белоусова

Содержание

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА	4
1.1. НОВИЗНА ДООП.....	4
1.2. АКТУАЛЬНОСТЬ ДООП.....	4
1.3. ПЕДАГОГИЧЕСКАЯ ЦЕЛЕСООБРАЗНОСТЬ	4
1.4. ЦЕЛЬ ДООП	5
1.5. ВОЗРАСТ ОБУЧАЮЩИХСЯ, УЧАСТВУЮЩИХ В ДООП.....	5
1.6. УСЛОВИЯ ВХОЖДЕНИЯ В ДООП	5
1.7. СРОК РЕАЛИЗАЦИИ ДООП.....	5
1.8. РЕЖИМ ЗАНЯТИЙ, ФОРМЫ И МЕТОДЫ ОБУЧЕНИЯ	6
1.9. ОЖИДАЕМЫЕ ОБРАЗОВАТЕЛЬНЫЕ РЕЗУЛЬТАТЫ И ЭФФЕКТЫ, СПОСОБЫ ПРЕДЪЯВЛЕНИЯ И ОТСЛЕЖИВАНИЯ РЕЗУЛЬТАТОВ	6
2. УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН.....	8
3. СОДЕРЖАНИЕ ДООП	9
4. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ И ИНФОРМАЦИОННО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ.....	12
5. РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА	13
6. СВЕДЕНИЯ О СОСТАВИТЕЛЯХ ДООП.....	13

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

В текущий момент, на основе анализа различных сервисов по поиску работы, спрос на вакансию разработчика мобильного программного превышает предложение. Мобильные устройства широко распространены и присутствуют практически во всех сферах деятельности общества, что дает широкий выбор направлений разработки.

Java является самым популярным языком программирования в течении последних лет, а благодаря своим особенностям, может быть использована не только для разработки мобильных приложений, но и для разработки программного обеспечения для персональных компьютеров и любых устройствах, где может быть запущена исполняющая среда. В настоящее время на большинстве мобильных устройств используется операционная система Android. Наиболее популярным языком для создания приложений под эту операционную систему является Java. Изучение языка программирования Java в рамках данной программы в основном подразумевает разработку мобильных приложений, работу со средами разработки, работу с различными медиа-контентом.

ДООП «Мобильная разработка» имеет техническую направленность, закладывает основы к дальнейшему развитию учащихся в области мобильной разработки, ориентирована на развитие навыков программирования и проектирования программ под платформу Android.

1.1. НОВИЗНА ДООП

Особенность ДООП «Мобильная разработка» – в изучении основ языка программирования Java и структуры приложения под операционную систему Android. Она строится в доступной и понятной для учащихся среде. Программирование ведется в текстово-графическом режиме, что позволяет постепенно наращивать необходимый функционал и получать необходимые инструменты для построения приложений.

1.2. АКТУАЛЬНОСТЬ ДООП

Актуальность данной ДООП состоит в том, что она составлена с учетом современных потребностей рынка в специалистах в области информационных технологий. Направление мобильной разработки - в списке ключевых и оказывает существенное влияние на развитие рынков НТИ (Национальной технологической инициативы).

Процесс усвоения нового учащимися происходит на практике. При этом каждый раздел курса содержит актуализированные теоретические материалы, необходимые для выполнения практических заданий.

Данная ДООП позволяет обучающимся самостоятельно выбрать актуальную проблемную область и создать проекты, конечные результаты которых будут полноценными техническими и творческими решениями или же могут быть продолжены и расширены в дальнейшем.

1.3. ПЕДАГОГИЧЕСКАЯ ЦЕЛЕСООБРАЗНОСТЬ

ДООП составлена в виде модулей, позволяющих обучающимся в максимально короткие сроки получить необходимый объем знаний и закрепить их на практике. Программа развивает логическое и алгоритмическое мышление, активизирует интерес к научно-техническому творчеству, одновременно развивая практические навыки в конкретной области мобильной разработки и «мягкие» навыки, необходимые для

успешной социализации в обществе. Учащиеся освоят основы языка программирования Java и получат востребованные навыки разработки приложений для платформы Android.

1.4. ЦЕЛЬ ДООП

Сформировать навыки программирования на языке Java и разработки мобильных приложений под операционную систему Android для реализации собственных проектов, вызванных конкретной потребностью каждого обучающегося.

Обучающие задачи:

- изучение основных понятий об алгоритмах;
- изучение алгоритма разработки проекта мобильного приложения;
- изучение современных и часто используемых платформ для разработки мобильных приложений;
 - формирование навыков объектно-ориентированного проектирования при разработке мобильного приложения;
 - формирование навыков программирования на языке Java;
 - изучение архитектуры мобильного приложения для операционной системы Android.

Развивающие задачи:

- расширение кругозора обучающихся в области программирования;
- формирование самостоятельности и творческого подхода к решению задач с использованием средств вычислительной техники;
 - развитие навыков поиска необходимой информации в сети Интернет, анализа выбранной информации и ее использования при решении задач;
 - развитие логического, технического и алгоритмического мышления;
 - развитие умения самостоятельно решать поставленную задачу, четко излагать мысли, анализировать ситуацию и самостоятельно находить ответы.

Воспитательные задачи:

- развить опыт групповой работы и отношений делового сотрудничества;
- создать условия для развития устойчивой потребности в самообразовании.

1.5. ВОЗРАСТ ОБУЧАЮЩИХСЯ, УЧАСТВУЮЩИХ В ДООП

Программа рассчитана на обучающихся 12-17 лет.

1.6. УСЛОВИЯ ВХОЖДЕНИЯ В ДООП

Набор на ДООП осуществляется в соответствии с Порядком приема и отчисления обучающихся Филиала АНО ДТ «Красноярский «Кванториум» в г. Норильске «Центр цифрового образования детей IT-Куб г. Норильск».

Поступающий на программу должен владеть базовыми навыками работы на компьютере (создание и удаление файлов; умение работать в простейшем текстовом и графическом редакторе; запуск, просмотр презентаций и видеороликов).

1.7. СРОК РЕАЛИЗАЦИИ ДООП

1 семестр. Нагрузка на обучающегося составляет 72 часа.

1.8. РЕЖИМ ЗАНЯТИЙ, ФОРМЫ И МЕТОДЫ ОБУЧЕНИЯ

Учебные занятия проходят по очной форме обучения. Режим занятий: 2 раза в неделю по 2 академических часа (1 академический час – 45 минут) с обязательным перерывом, что определяется Санитарно-эпидемиологическими правилами и нормативами СанПиН 2.4.4.3172-14.

При проведении занятий используются комбинированные занятия – изложение нового материала, проверка пройденного материала, закрепление полученных знаний, самостоятельная работа.

При проведении занятий используются следующие формы работы:

- демонстрационная, когда обучающиеся слушают объяснения педагога и наблюдают за демонстрационным экраном или экранами компьютеров на ученических рабочих местах;

- фронтальная, когда обучающиеся синхронно работают под управлением педагога;

- самостоятельная, когда обучающиеся выполняют индивидуальные задания в течение части занятия.

Повторение и усвоение пройденного материала осуществляется через контрольные и проверочные работы, анализ полученных результатов.

Закрепление знаний, умений и навыков осуществляется через постановку задачи и самостоятельную работу обучающегося под руководством педагога.

Применение полученных знаний и навыков осуществляется через прикладную работу обучающегося, использующего на практике приобретенные компетенции.

1.9. ОЖИДАЕМЫЕ ОБРАЗОВАТЕЛЬНЫЕ РЕЗУЛЬТАТЫ И ЭФФЕКТЫ, СПОСОБЫ ПРЕДЪЯВЛЕНИЯ И ОТСЛЕЖИВАНИЯ РЕЗУЛЬТАТОВ

По результатам освоения программы, обучающиеся овладевают базовыми основами технологии программирования на языке Java для создания собственных проектов под платформу Android и должны:

Уметь:

- пользоваться готовыми компонентами для разработки приложений;
- разрабатывать и использовать алгоритмы;
- находить в сети Интернет необходимую информацию;
- разрабатывать мобильное приложение минимального функционала;
- пользоваться ПК и IDE-разработкой для программирования устройства;
- читать готовую программу и находить ошибки в готовых программах.

Знать:

- основные приемы написания программ-приложений;
- возможности визуального программирования;
- требования к написанию и оформлению программ-приложений;
- основные алгоритмические конструкции;
- простейшие типы данных и основные операторы языка программирования;
- понятия «функции», «параметры», «глобальные и локальные переменные»;
- основы языка программирования Java и языка разметки XML;
- принцип работы баз данных и клиент-серверных протоколов;

- методы производства и проектирования мобильных приложений.

Обладать навыками:

- соблюдения требований техники безопасности;
- работы в изучаемых программных средах;
- составления алгоритмов;
- применения на практике основных команд и операторов изучаемых языков;
- разработки, тестирования и отладки несложных программ.

Механизм оценки результативности

Контроль степени освоения программы осуществляется в форме тестов, выполнения практических и проектных работ. Пример проверочных тестов представлен в приложении 1.

Тесты и практические работы по каждому разделу оцениваются по 5 баллам. Их содержание соответствует темам модулей, а содержание определяется наставником. Проект и презентация проекта оцениваются отдельно по трёхбалльной шкале, с последующими рекомендациями и замечаниями при наличии. Темы проектов, учащиеся формируют в процессе обучения и корректируют с наставником. При возникновении осложнений с выбором тем, учащимся может быть предоставлен рекомендованный список тем (Приложение №2).

По итогам каждого этапа проводится аттестация в форме подготовки небольшой проектной работы, являющаяся промежуточной аттестацией по программе.

2. УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН

№	Наименование разделов	Всего часов	Теория	Практика	Форма контроля
1	Знакомство с направлением обучения	2	2	0	
1.1.	Вводный урок. Правила работы и техника безопасности при работе на компьютере.	1	1	0	
1.2.	Структура курса. Ожидаемые результаты.	1	1	0	
2	Основы программирования	14	6	8	
2.1.	Знакомство с Java. Среда разработки. Типы данных. Математические операции.	6	3	3	
2.2.	Алгоритмические, условные и итеративные конструкции. Методы.	6	3	3	
2.3.	Практическая работа. Тестирование полученных знаний.	2	0	2	Тест
3	Объектно-ориентированное программирование	14	5	9	
3.1.	Классы, объекты. Наследование. Полиморфизм.	8	3	5	
3.2.	Архитектура приложений Android. Интерфейс пользователя. Разметка XML.	4	2	2	
3.3.	Практическая работа. Тестирование полученных знаний.	2	0	2	Тест
4	Проектирование Android приложений	12	5	7	
4.1.	Ввод. Вывод. Отладка. Потоки.	6	3	3	
4.2.	Графика в Android приложениях. Разработка графики.	4	2	2	
4.3.	Практическая работа. Тестирование полученных знаний.	2	0	2	Тест
5	Структуры данных. Алгоритмы.	14	6	8	
5.1.	Структуры данных. Адаптеры. Хэш.	4	2	2	
5.2.	Рекурсия. Алгоритмы. Поиск. Сортировка.	4	2	2	
5.3.	Базы данных.	4	2	2	
5.4.	Практическая работа. Тестирование полученных знаний.	2	0	2	Тест
6	Серверная часть мобильных приложений	10	4	6	
6.1.	Сеть и серверы. Работа с HTTP.	4	2	2	
6.2.	Клиент-серверная архитектура. Облака.	4	2	2	
6.3.	Практическая работа. Тестирование полученных знаний.	2	0	2	Тест
7	Доработка и сдача проектов	6	0	6	
ИТОГ		72	28	44	

3. СОДЕРЖАНИЕ ДООП

ТЕМА 1. Знакомство с направлением обучения

1.1. Вводный урок. Правила работы и техника безопасности при работе на компьютере.

Теоретическая работа: Правила работы в помещении. Техника безопасности при работе за компьютером.

1.2. Структура курса. Ожидаемые результаты.

Теоретическая работа: Структура курса. Ожидаемые результаты обеих сторон. План работы.

ТЕМА 2. Основы программирования

2.1. Знакомство с Java. Среда разработки. Типы данных. Математические операции.

Теоретическая работа: Среда разработки IntelliJ IDEA. Понятие проекта. Порядок создания, компиляции, сборки и запуска приложения. Порядок установки среды разработки на домашнем компьютере. Понятие пакета (package). Роль метода main(). Комментарии. Демонстрация типичных ошибок, возникающих при компиляции и выполнении Java-программ. Переменные и константы. Тип переменной. Final. Целочисленные типы данных. Разновидности типа «целое»: int, short, long, byte. Типы с плавающей точкой. Типы float, double. Char. Арифметические выражения и операторы, операторы присваивания. Сложение, вычитание, умножение, деление, остаток от деления (%), инкремент (++), декремент (--). Префиксная и постфиксная запись инкремента и декремента. Скобки. Операторы присваивания (=, +=, -= и т.д.). Порядок исполнения. Кодировки (UTF-8, 16, Unicode). Тип данных boolean. Логические значения true и false. Логические операции и операции отношения. Операторы отношения: >, <, >=, <=, !=, ==. Тернарная операция «?:».

Практическая работа: Разбор Java-проекта. Имя файла=имя проекта. Написание и отладка программы, выводящей информацию об ученике. Вывод и ввод различных типов данных. Полномка компилятора и ввода. Перевод типов данных из одного в другой. Программа для определения типа введенных данных. Тестовые программы для уяснения порядка выполнения операций. Переписывание текстовых логических задач в программы.

2.2. Алгоритмические, условные и итеративные конструкции. Методы.

Теоретическая работа: Область действия блоков. Фигурные скобки. Вложенность блоков. Statement. «if-else:». Конструкция switch-case. Синтаксис. Сложные ветвления. break и default в конструкции switch-case. for, for each, одномерные массивы. Цикл с предусловием while, примеры. Цикл с постусловием do-while, примеры. Безусловные операторы перехода break, continue. Массивы. Цикл for. Частные формы записи оператора for. Примеры некорректного использования. Цикл for each. Понятие функции на примере модулей. Определение функции. Void. Область видимости переменных. Матрицы. Отсутствие двумерных массивов. Использование вложенного цикла for each.

Практическая работа: Программы со сложным ветвлением, оптимизации прошлых итерационных задач. Нахождение максимума/минимума трех чисел. Программы с использованием циклов. Работа с массивами. Примеры с необходимым использованием функций. Собственные функции работы с массивами. Объявление и обработка матриц/неровных массивов. Генерация sudoku.

2.3. Практическая работа. Тестирование полученных знаний.

Практическая работа: Практическое занятие, тестирование по темам модуля, для закрепления материала.

ТЕМА 3. Объектно-ориентированное программирование

3.1. Классы, объекты. Наследование. Полиморфизм.

Теоретическая работа: Основные понятия. Класс, объект (экземпляр класса), переменные (поля) класса, метод класса. ОО-подход как естественное отражение мира. Описание протокола класса. Ключевое слово `class`. Поля класса. Метод класса, его аргументы и возвращаемое значение, описание в протоколе. `new`. Вызов метода через переменную – объект собственного класса. Интерпретация метода как посылки сообщения объекту. Инкапсуляция, наследование, полиморфизм. Конструкторы и деструкторы в Java и их использование. Перегрузка методов. Ключевое слово `this`. Спецификаторы доступа. Защищенные члены класса. Наследование. Расширение базового класса через наследников. Примеры наследования. Интерфейс. Базовый класс. `extends` и `implements`. `super`. `final`. Применение `final`. Полиморфизм как отделение интерфейса от реализации. Позднее связывание и связь с наследованием. Абстрактные классы и методы.

Практическая работа: Проектирование и реализация простейшего класса. Знакомство с классами примитивного типа. Генерирование примеров классов и соответствующих им примеров объектов, полей и методов. Пример перегрузки конструктора и обычного метода. Продолжение разработки собственного класса. Реализация конструкторов. Пример описания классов, наследования, переопределение метода, управление доступом, базовый тип. Объект производного класса является экземпляром базового класса. Создание собственного класса по аналогии с широко известным объектом мира или математики. Пример замены наследования вложенностью классов. Пример с циклическим вызовом одного метода с разной реализацией. Полиморфизм и абстракция на примере метода отрисовки. Отрисовка фигур.

3.2. Архитектура приложений Android. Интерфейс пользователя. Разметка XML.

Теоретическая работа: Android Studio. Общая структурная схема приложения под Android. Жизненный цикл Android-приложения. `Activity`, создание, жизненный цикл, стеки, состояние. Структура XML. `Layouts`. `Views`. Контекст. `Intents`. Явные и неявные намерения.

Практическая работа: Установка IDE и эмулятора для разработки приложений под Android на ПК. Разбор и комментирование схемы (используется схема из `developer.android.com`). Разбор простого приложения под Android. Модификация этого приложения. Примеры использования XML. Применение `Layouts` и `Views`. Создать описание «рюкзака» с помощью языка XML. Прогресс бар для процесса вычисления. Создание нескольких `Activity` (и экранов) в одном приложении.

3.3. Практическая работа. Тестирование полученных знаний.

Практическая работа: Практическое занятие, тестирование по темам модуля, для закрепления материала.

ТЕМА 4. Проектирование Android приложений

4.1. Ввод. Вывод. Отладка. Поток.

Теоретическая работа: Класс File и его методы. Обработка исключения с помощью конструкции try-catch. Exception. Прерывание и возобновление. Внутренние (вложенные) классы. Listeners. Использование анонимных классов для реализации обработчиков событий. Параллелизм и синхронизация. Поток. Threads. AsyncTask. Синхронизация потоков. Блокировки. Deadlock.

Практическая работа: Примеры возникновения и обработки исключений, возникающих при выполнении методов класса. Использование классов Scanner и PrintWriter. Разбор примера кода с обработчиками событий. Пример нарушения целостности данных при неаккуратном параллелизме.

4.2. Графика в Android приложениях. Разработка графики.

Теоретическая работа: Класс Canvas. Класс Paint. Двухмерная анимация. Класс SurfaceView. Отличие View от SurfaceView. Игры.

Практическая работа. Простой визуальный пользовательский интерфейс. Разбор примера простейшей игры с анимацией.

4.3. Практическая работа. Тестирование полученных знаний.

Практическая работа: Практическое занятие, тестирование по темам модуля, для закрепления материала.

ТЕМА 5. Структуры данных. Алгоритмы.

5.1. Структуры данных. Адаптеры. Хэш.

Теоретическая работа: Arrays, ArrayList. Итератор. Обход контейнеров. Понятие Список. Создание списков. LIFO. FIFO. LinkedList, Queue, Stack. Адаптеры. Назначение адаптеров. Метод hashCode().

Практическая работа: Работа с библиотечным классом Arrays. Пример применения классов LinkedList, Queue и Stack. Реализация ограниченной очереди. Отображение ListView через адаптер. Пример реализации метода hashCode().

5.2. Рекурсия. Алгоритмы. Поиск. Сортировка.

Теоретическая работа: Значение алгоритмов сортировки. Быстрая сортировка. Поиск. Линейная и ветвящаяся рекурсия.

Практическая работа: Поиск методами Arrays. Реализация функции, проверяющей, является ли массив отсортированным. Пример рекурсии.

5.3. Базы данных.

Теоретическая работа: Хранение данных в таблицах. Связи и отношения. Локальные СУБД. Введение в SQL. Локальная СУБД SQLite. Язык запросов SQL. Обновление таблиц. Выборка данных. Булевы операции. Агрегация.

Практическая работа: Практика с базами данных. Создание Записной книжки.

5.4. Практическая работа. Тестирование полученных знаний.

Практическая работа. Практическое занятие, тестирование по темам модуля для закрепления материала.

ТЕМА 6. Серверная часть мобильных приложений.

6.1. Сеть и серверы. Работа с HTTP.

Теоретическая работа: Интернет и протоколы TCP/IP. Команды ping, tracert, nslookup, ipconfig. Структура HTTP-запроса. Web-сервер. Методы GET и POST. Ответы сервера.

Практическая работа: Определение собственного внешнего IP-адреса. Сервис whois. Аprobация команд. Примеры запросов GET и POST.Dev tools в браузере.

6.2. Клиент-серверная архитектура. Облака.

Теоретическая работа: Структура и схема взаимодействия в клиент-серверных мобильных приложениях. Синхронные и асинхронные запросы. Формат JSON и XML. Сериализация. Retrofit. Облачные платформы. Heroku. Основные методы HTTP-запроса. OAuth на примере VK.

Практическая работа: Отправка запросов из Android-приложений. Приложение для отправки данных на сервер. Использование AsyncTask. Предиктор Яндекса. Создание клиент-серверного приложения, позволяющего авторизоваться через OAuth-авторизацию и отправлять информацию о пользователе на сервер.

6.3. Практическая работа. Тестирование полученных знаний.

Практическая работа: Практическое занятие, тестирование по темам модуля, для закрепления материала.

ТЕМА 7. Доработка и сдача проектов.

В рамках данного модуля учащиеся получают рекомендации, дорабатывают и сдают свои проекты с презентацией.

4. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ И ИНФОРМАЦИОННО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

Помещения для лабораторных занятий и самостоятельной работы обучающихся должны быть оснащены персональными компьютерами с установленной средой разработки программного обеспечения на языке объектно-ориентированного программирования Microsoft Visual Studio Community 2019 (и выше), IDE PyCharm 2018.3.5 (и выше), Python 3.7.4 (и выше), Visual Studio Code 1.35.0 (и выше), расширение для VSCode - Live Server, расширение для VSCode - CSS Peak, расширение для VSCode - Prettier, Microsoft Office 2013 (и выше), с возможностью подключения к сети Интернет и доской с маркером или интерактивной доской.

Помещение для занятий лекционного типа должно быть оснащено мультимедийным проектором и персональными компьютерами с установленной средой разработки программного обеспечения на языке объектно-ориентированного программирования Microsoft Visual Studio Community 2019 (и выше), Microsoft Office 2013 (и выше), с возможностью подключения к сети Интернет, доской с маркером или интерактивной доской.

Перечень программного обеспечения: IDE IntelliJ idea; Android Studio; Paint.NET.

5. РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

Основная литература:

1. Шилдт Герберт «Java. Полное руководство». М: Диалектика, 2018 г.
2. Шилдт Герберт «Java. Руководство для начинающих». М.: Диалектика, 2018 г.
3. Эккель Брюс «Философия Java». М: Питер, 2019 г.

Дополнительная литература:

1. Блох Джошуа. Java. Эффективное программирование. Effective Java. Programming Language Guide. М.: Лори, 2014 г.
2. Гослинг Джеймс, Билл Джой, Гай Л. Стил, Гилад Брача, Алекс Бакли. Язык программирования Java SE 8. Подробное описание. The Java Language Specification: Java SE8 Edition. М.: Вильямс, 2015.
3. Зигард Медникс, Лайрд Дорнин, Блейк Мик, Масуми Накамура. Программирование под Android. Programming Android. СПб.: Питер, 2012.
4. Майер Рето. Android 2. Программирование приложений для планшетных компьютеров и смартфонов. Professional Android 2: Application Development 2nd Edition. М.: Эксмо, 2011.
5. Файн Яков. Программирование на Java для детей, родителей, дедушек и бабушек. Самиздат, 2011.
6. Харди Брайн, Билл Филлипс. Программирование под Android. Android Programming: The Big Nerd Ranch Guide. СПб.: Питер, 2014.

Электронные ресурсы:

1. Сайт developer.android.com – информация по Android разработке и техническая документация.
2. Сайт java.com – техническая документация и обновления Java
3. Сайт material.io – guidelines по дизайну мобильных приложений.

6. СВЕДЕНИЯ О СОСТАВИТЕЛЯХ ДООП

В.М. Поляков педагог дополнительного образования, преподаватель по направлениям «Мобильная разработка». Образование высшее. ФГБОУ ВО «Норильский государственный индустриальный институт», диплом по направлению «Информационные системы и технологии», квалификация «Бакалавр».

1. Что такое ООП?

a. Объектно-ориентированное программирование — методология программирования, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определенного класса, а классы образуют иерархию наследования.

b. Объектно-ориентированное программирование — так называют любой тип программирования, в котором используются понятия высокого уровня и, в отличие от Assembler, в котором не работают напрямую с ячейками памяти ПК.

c. Объектно-ориентированное программирование — просто красивое понятие. Если вдуматься, оно не несет дополнительной смысловой нагрузки, просто программисты любят аббревиатуры, так области их знаний выглядят сложнее.

d. Очень одинокий программист.

2. Что такое класс в Java?

a. Уровень сложности программы. Все операторы делятся на классы в зависимости от сложности их использования.

b. Базовый элемент объектно-ориентированного программирования в языке Java.

c. Просто одно из возможных названий переменной.

d. Такое понятие есть только в C++, в Java такого понятия нет.

3. Как объявить класс в коде?

a. `class MyClass { }`

b. `new class MyClass { }`

c. `select * from class MyClass { }`

d. `MyClass extends class { }`

4. Что выведется в результате этого кода?

```
int a = 5;
int b = 5+a;
if (a > b) {
    System.out.println("a > b");
} else if (a < b) {
    System.out.println("a < b");
} else {
    System.out.println("a = b");
}
```

a. Код написан с ошибкой, ничего не выведется.

b. `a > b`

c. `a = b`

d. `a < b`

5. Для чего используется оператор NEW?

a. Для создания новой переменной.

b. Для объявления нового класса.

c. Для создания экземпляра класса.

d. Это антагонист оператора OLD.

6. Как объявить класс в коде?

```
String a = "my test";  
String b = "this is";  
String c;  
  
switch (a) {  
    case "MY TEST":  
        c = a + b;  
        break;  
  
    case "My test":  
        c = b + a;  
        break;  
  
    case "MY TEST":  
        c = a + b;  
        break;  
  
    case "my_test":  
        c = a + a;  
        break;  
    default:  
        c = a + a;  
}  
System.out.println(c);
```

- a. this is my test
- b. this isthis is
- c. my testmy test
- d. my test this is

7. Что означает ключевое слово `extends`?

- a. Что данный класс наследуется от другого.
- b. Что это дополнительный модуль класса, который расширяет его свойства.
- c. Что два класса делают одно и то же.
- d. Что это самый большой класс в программе.

8. Что означает ошибка `Exception <...>`

`Java.lang.ArrayIndexOutOfBoundsException: 4?`

- a. Ошибка означает переполнение памяти.
- b. Ошибка означает попытку вставить в массив некорректное значение.
- c. Ошибка означает выход за рамки границ массива.
- d. Такой ошибки нет.

9. Что означает перегрузка метода в Java (`overload`)?

- a. Изменение поведения метода класса относительно родительского.
- b. Изменение поведения метода класса относительно дочернего.
- c. Несколько разных классов с одинаковым методом.
- d. Несколько методов с одинаковым названием, но разным набором параметров.

10. Что означает переопределение метода в Java (override)?

- a. Несколько методов с одинаковым названием, но разным набором параметров.
- b. Изменение поведения метода класса относительно родительского.
- c. Несколько разных классов с одинаковым методом.
- d. Изменение поведения метода класса относительно дочернего.

11. Чем отличаются static-метод класса от обычного метода класса?

- a. Обычный метод класса можно переопределить, а static-метод нельзя.
- b. Обычный метод класса работает от объекта класса, а static-метод от всего класса.
- c. Static-метод класса можно вызывать только внутри класса, а обычный - в любой части кода.
- d. Обычный метод класса можно перегрузить, а static-метод нельзя.

12. Для чего необходимо ключевое слово this

- a. Это указатель на переопределенный метод класса. Его нельзя опускать при вызове, иначе переопределение не сработает.
- b. Это не ключевое слово.
- c. Это ключевое слово для вызова обычного метода внутри static-метода. Его нельзя опускать, иначе вызов не сработает и будет ошибка.
- d. Это указатель на текущий объект класса внутри самого класса. Его можно опускать при вызове метода класса, но лучше этого не делать.

13. Что вернет метод, объявленный следующим образом:

public static int getAmount()?

- a. Вернет static-поле класса.
- b. Вернет ссылку на объект класса this.
- c. Вернет целочисленное значение.
- d. Не ясно, надо посмотреть код метода.

14. Как исправить этот код, чтобы он вывел "I Love To Learn" корректно?

```
String[] text = new String[] {  
    "I",  
    "Love",  
    "To",  
    "Learn"  
};  
int i = 1;  
while (I < text.length) {  
    System.out.print(i);  
}
```

- a. Надо внутри while добавить i++ в конце, иначе будет бесконечный цикл.
- b. Надо в System.out.print передавать text[i].
- c. Надо объявить int i = 0.
- d. Надо выполнить все предыдущие пункты для корректной работы программы.

Список рекомендованных тем:

1. Формирование счета-фактуры.
2. AR-игра.
3. Текстовый интерактивный квест.
4. Справочник по узким направлениям.
5. Бронирование столов/номеров.
6. Приложение для знакомств.
7. Записная книжка.
8. Отслеживание задач.
9. Управление умным домом.
10. Поиск людей с необходимым навыком.
11. Организация доставки.
12. Архитектурный AR-справочник.
13. Шар судьбы.
14. Дайджест мероприятий.
15. Общественный сборник слухов.
16. Сортировка мусора.
17. Локальный сервис для попутчиков.
18. Файловый менеджер.
19. Учет для склада/кладовки.