

Министерство образования и науки Хабаровского края
Краевое государственное автономное образовательное учреждение дополнительного
образования «Центр развития творчества детей (Региональный модельный центр
дополнительного образования детей Хабаровского края)»
Центр технического и цифрового образования «ТЕХНО-IT-клуб»
наименование структурного подразделения

Рассмотрена

на заседании научно-
методического совета Центра

Протокол № 3

«30» 06 2023 г.

Утверждаю

Генеральный директор
КГАОУ ДО РМЦ



М.В. Кацупий

«30» 06 2023 г.

**ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА
ТЕХНИЧЕСКОЙ НАПРАВЛЕННОСТИ**

«Мобильная разработка»

название ДООП

Возраст учащихся: 14-18 лет

Срок реализации: 1 год

Уровень освоения: продвинутый

Составитель: Усов Сергей
Константинович, педагог
дополнительного образования

г. Хабаровск,
2023 г.

1. Комплекс основных характеристик ДООП

1.1. Пояснительная записка

Программа разработана с учетом следующих нормативно-правовых документов:

1. Федеральный Закон от 29.12.2012г. № 273-ФЗ «Об образовании в Российской Федерации».

2. Приказ Министерства просвещения Российской Федерации от 27.07.2022 г. № 629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам».

3. Постановление Главного государственного санитарного врача РФ от 28 сентября 2020 г. N 28 "Об утверждении санитарных правил СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи».

4. Распоряжение Правительства Российской Федерации от 31.03.2022г. № 678-р «Концепция развития дополнительного образования детей до 2030 года».

5. Устав краевого государственного автономного образовательного учреждения дополнительного образования «Центр развития творчества детей (Региональный модельный центр дополнительного образования детей Хабаровского края)».

6. Протокол заочного заседания Рабочей группы по дополнительному образованию детей Экспертного совета Министерства просвещения Российской Федерации по вопросам дополнительного образования детей и взрослых, воспитания и детского отдыха от 22 марта 2023 г. № Д06-23/06пр.

Перечень дополнительной документации:

1. Программа, разработанная Samsung R&D Institute Rus (Исследовательский Центр Samsung) при участии Московского физико-технического института (МФТИ) для IT школа Samsung

Направленность программы: техническая

Уровень сложности содержания программы: продвинутый (1 год).

Актуальность: Обусловлена широким распространением мобильных устройств, что делает востребованной профессию мобильного разработчика. Программа продвинутого уровня поможет учащимся освоить фундаментальные разделы ИТ и программирования в адаптированной для школьников форме, получить практические навыки по разработке мобильных приложений в идеологии промышленной индустрии «от идеи до дистрибуции». Изучение программы подойдет для будущих программистов, инженеров, дизайнеров и разработчиков интерфейсов, ученых, техников.

Педагогическая целесообразность: Занятия по данной дополнительной образовательной программе смогут помочь обучающимся,

выявить свои интересы в сфере информационных технологий и способствовать профессиональной ориентации.

По результатам успешного окончания программы и защиты проекта учащимся вручается сертификат от компании Samsung, который засчитывается в виде дополнительных баллов при зачислении на бюджетной основе в 19 вузах России.

Новизна программы: Комплексный подход к обучению программирования приложений для платформы android, возможность дистанционного изучения дисциплины, широкая направленность. Программа составлена с учетом текущих требований к разработке приложений в практической деятельности. Материалы программы дублируются на сайте IT-школы Samsung (<https://myitschool.ru>), что обеспечивает возможность дистанционного изучения обучающих материалов.

Адресат программы: Возраст обучающихся: 14-18 лет;

Программа ориентирована на учащихся, обладающими следующими качествами:

- имеют склонность к алгоритмическому мышлению, увлекаются ИТ-технологиями;
- владеют хотя бы одним языком программирования на уровне знания условных и циклических конструкций;
- имеют устойчивые знания по школьному курсу математики.

Отбор детей происходит по итогам тестирования партнера программы <https://www.samsung.com/ru/itschool/about/>.

Форма обучения: очная

Срок реализации программы: 1 год.

Объём реализации программы: 180 часов – «ТЕХНО-ИТ-куб»

Период	Продолжительность занятия	Кол-во занятий в неделю	Кол-во часов в неделю	Кол-во недель	Кол-во часов в год
1 год обучения	2 часа	3	5 часов	36	180 часов

Режим организации занятий: Занятия в объединении проходят 3 раза в неделю: 2 раза по 2 часа и 1 раз по 1 часу. Занятия проводятся по 45 минут. Между занятиями 10-минутный перерыв.

Форма организации занятий: основная форма организации занятий – групповые, практические занятия. Группы должны состоять из 7 - 12 человек.

1.2.Цели и задачи программы

Цель: Развитие у обучающихся информационно-коммуникационных компетенций на углубленном уровне.

Задачи программы:

- Способствовать формированию устойчивого интереса к информационным технологиям;
- Формировать потребность в творческой деятельности в области разработки цифровых продуктов;

– Способствовать применению коммуникативных навыков в совместной творческо-продуктивной деятельности.

1.3. Учебный план

№ п/п	Название раздела, темы	Количество часов			Формы аттестации (контроля)
		всего	теория	практика	
Раздел 1. Основы программирования Java		30	10	20	
1.1	Знакомство с системой обучения в ИТ ШКОЛЕ SAMSUNG. Порядок создания, компиляции, сборки и запуска в IDE приложения Java	2	2		Устный опрос
1.2	Типы данных и операции	2		2	Практическая работа
1.3	Представление целочисленных данных в памяти	2		2	Практическая работа
1.4	Представление отрицательных и вещественных чисел. Поразрядные операции	4	2	2	Устный опрос
1.5	Логические выражения	2		2	Практическая работа
1.6	Условный оператор	2		2	Практическая работа
1.7	Циклы while, do while	2		2	Практическая работа
1.8	Цикл for. Массивы	4	2	2	Устный опрос
1.9	Работа с циклами	2		2	Практическая работа
1.10	Методы (процедуры и функции)	2		2	Практическая работа
1.11	Многомерные массивы	2	2		Устный опрос
1.12	Неровные массивы	2	2		Устный опрос
1.13	Контрольное тестирование по модулю	2		2	Практическая работа
Раздел 2. Введение в объектно-ориентированное программирование		34	8	26	
2.1	Классы и объекты	4	2	2	Опрос
2.2	Конструкторы и их перегрузка	2		2	Практическая работа
2.3	Статические поля и методы	2		2	Практическая работа
2.4	Приемы тестирования и отладки на примерах со строками	2		2	Практическая работа
2.5	Знакомство с Android-разработкой	4	2	2	Опрос
2.6	Интерфейс Android-приложения	2		2	Практическая работа
2.7	Наследование и полиморфизм в Java	4	2	2	Опрос
2.8	Context и Intent	2		2	Практическая работа
2.9	Параметризованные типы	2		2	Практическая работа

2.10	Классы и объекты. Конструкторы. Ключевое слово «static»	2		2	Практическая работа
2.11	Уровни доступа	2		2	Практическая работа
2.12	XML разметка	4	2	2	Опрос
2.13	Контрольное тестирование по модулю	2		2	Практическая работа
Раздел 3. Основы программирования Android-приложений		36	8	28	
3.1	Объектно-ориентированное проектирование	4	2	2	Опрос
3.2	Ввод, вывод и исключения	2		2	Практическая работа
3.3	Внутренние и анонимные классы	2		2	Практическая работа
3.4	Параллелизм и синхронизация, потоки	2	2		Опрос
3.5	Сервисы в Android. Типы сенсоров	2		2	Практическая работа
3.6	Фрагменты в Android	2		2	Практическая работа
3.7	Двумерная графика в Android-приложениях	4	2	2	Опрос
3.8	Двумерная графика в Android-приложениях	2		2	Практическая работа
3.9	Разработка игровых приложений. SurfaceView	2		2	Практическая работа
3.10	Разработка игровых приложений. SurfaceView	2		2	Практическая работа
3.11	Разработка 3D-игр с использованием фреймворка libGDX	4	2	2	Опрос
3.12	Практикум	2		2	Практическая работа
3.13	Защита прототипа индивидуального проекта	2		2	Практическая работа
3.14	Защита прототипа индивидуального проекта	2		2	Практическая работа
3.15	Контрольное тестирование по модулю	2		2	Практическая работа
Раздел 4. Алгоритмы и структуры данных		50	10	40	
4.1	Массивы. Класс ArrayList	2		2	Практическая работа
4.2	Связные списки. Очереди, стеки, деки	2		2	Практическая работа
4.3	Списки в Android. Адаптеры	2		2	Практическая работа
4.4	СУБД. Реляционная модель	2		2	Практическая работа
4.5	СУБД SQLite. Основы языка SQL	4	2	2	Опрос
4.6	Рекурсия	4	2	2	Опрос
4.7	Деревья	2		2	Практическая работа

4.8	Алгоритмы сортировок	2		2	Практическая работа
4.9	Множества. Хеширование	4	2	2	Опрос
4.10	Ассоциативные массивы	4	2	2	Опрос
4.11	Контент-провайдеры в Android	2		2	Практическая работа
4.12	Введение в криптографию и криптоанализ	2		2	Практическая работа
4.13	Ключи шифрования	2		2	Практическая работа
4.14	Дайджесты сообщений.	2		2	Практическая работа
4.15	Практическое занятие по библиотечному классу MessageDigest	2		2	Практическая работа
4.16	Электронно-цифровые подписи	2		2	Практическая работа
4.17	Практикум	2		2	Практическая работа
4.18	Предзащита индивидуального проекта	4	2	2	Опрос
4.19	Предзащита индивидуального проекта	2		2	Практическая работа
4.20	Контрольное тестирование	2		2	Практическая работа
Раздел 5. Основы разработки серверной части мобильных приложений		26	6	20	
5.1	IP-сети	2		2	Практическая работа
5.2	Веб-сервер, HTTP-запросы и ответы	4	2	2	Опрос
5.3	Клиент-серверная архитектура мобильных приложений	2		2	Практическая работа
5.4	Облачные платформы. REST-взаимодействие	4	2	2	Опрос
5.5	Серверные СУБД	4	2	2	Опрос
5.6	Практикум	2		2	Практическая работа
5.7	Практикум	2		2	Практическая работа
5.8	Контрольное тестирование	2		2	Практическая работа
5.9	Подготовка к защите итогового проекта	2		2	Практическая работа
5.10	Подготовка к защите итогового проекта	2		2	Практическая работа
Раздел 6. Итоговая аттестация		4		4	
6.1	Итоговый проект. Защита проектов	4	0	4	Устный опрос
	Итого	180	42	138	

1.4. Содержание учебного плана

Раздел 1. Основы программирования Java.

Теория: Типы данных и операции. Представление целочисленных данных в памяти. Представление отрицательных и вещественных чисел. Поразрядные операции.

Практика: Интерфейс IntelliJ IDEA. Создание проекта, структура программы на языке Java. Решение типовых задач.

Раздел 2. Введение в объектно-ориентированное программирование.

Теория: Классы и объекты. Конструкторы и их перегрузка. Статические поля и методы. Приемы тестирования и отладки на примерах со строками. Знакомство с Android-разработкой. Интерфейс Android-приложения. Наследование и полиморфизм в Java. Намерения. Параметризованные типы

Практика: Интерфейс Android studio. Структура проекта в Android studio. Решение задач. Индивидуальное задание.

Раздел 3. Основы программирования Android-приложений

Теория: Объектно-ориентированное проектирование. Ввод, вывод и исключения. Внутренние и анонимные классы Параллелизм и синхронизация, потоки. Сервисы в Android. Типы сенсоров. Фрагменты в Android. Двумерная графика в Android-приложениях.

Практика: Разработка игровых приложений. SurfaceView. Разработка 3D-игр с использованием фреймворка libGDX. Решение задач. Индивидуальное задание.

Раздел 4. Алгоритмы и структуры данных.

Теория: Массивы. Класс ArrayList. Связные списки. Очереди, стеки, деки. Списки в Android. Адаптеры. СУБД. Реляционная модель. Рекурсия. Деревья. Алгоритмы сортировок. Множества. Хеширование. Ассоциативные массивы. Введение в криптографию и криптоанализ. Ключи шифрования

Практика: СУБД SQLite. Основы языка SQL. Контент-провайдеры в Android. Решение задач.

Раздел 5. Основы разработки серверной части мобильных приложений.

Теория: IP-сети. Веб-сервер, HTTP-запросы и ответы. Клиент-серверная архитектура мобильных приложений. Облачные платформы. Серверные СУБД.

Практика: Дизайн программного обеспечения и приложений Material Design. REST-взаимодействие. Итоговый проект.

Раздел 6. Итоговая аттестация

Защита проекта.

1.5. Планируемые результаты

Предметные результаты:

- научиться разработке мобильных приложений для платформы Android;
- будут знать основы программирования на языке Java;
- смогут показать умения ведения проекта на различных этапах его жизненного цикла.

Метапредметные результаты:

- будут демонстрировать навыки планирования работы, самостоятельного выбора технологий, инструментов и форм для достижения поставленной задачи, цели;
- будут уметь самостоятельно контролировать ход выполнения работы, фиксировать последовательность и оценивать результат.

Личностные результаты:

- будут демонстрировать навыки самостоятельной и коллективной работы;
- будут демонстрировать навыки самоорганизации и планирования времени и ресурсов.

2. Комплекс организационно – педагогических условий

2.1. Календарный учебный график

Год обучения	Дата начала занятий	Дата окончания занятий	Кол-во учебных недель	Кол-во учебных дней	Кол-во учебных часов	Режим занятий
1 год обучения	01.09.2023г.	31.05.2024г.	36	108	180	2 раза в нед. по 2 часа 1 раз в нед. по 1 часу

2.2. Условия реализации программы

Материально-техническое обеспечение:

Требования к помещению:

- помещение для занятий, отвечающие требованиям СанПин для учреждений дополнительного образования;
- качественное освещение;
- столы, стулья по количеству обучающихся и 1 рабочим местом для педагога.

Оборудование:

- персональные компьютеры/ ноутбуки /планшетные компьютеры (по численности группы): • Intel Core i5 • 8 Гб ОЗУ • 120Гб SSD – 12 шт.;
- планшеты Samsung Tab A -12 шт;
- мультимедийный проектор или аналогичное оборудование для воспроизведения презентаций;
- доступ в сеть интернет не менее 10 Мбит/с на класс;

Программное обеспечение:

- операционная система Windows;
- браузер Google Chrome;

– программное обеспечение IntelliJ IDEA, Android studio.

Кадровое обеспечение:

Программа реализуется педагогом дополнительного образования, имеющего аттестацию по программе Samsung

Информационное обеспечение:

Материалы IT-школы Samsung. Web-сайт IT-школы Samsung www.myitschool.ru

2.3. Формы аттестации

По итогам освоения раздела в рамках программы предусмотрено тестирования ученика на предмет качества усвоения материала.

По итогам учебного года в рамках образовательной программы предусмотрено выполнение итогового проекта обучающимися с целью отображения качества усвоения программы в рамках практической деятельности.

Формы представления результатов - представление проектов на открытом занятии с приглашением родителей.

2.4. Оценочные материалы

Тестовые задания по темам на платформе IT-школы Samsung (Приложение 1).

Общими критериями оценки итоговых проектов обучающихся являются:

- Идея проекта, функциональная полнота (0-5 баллов);
- Работоспособность, техническая реализация (0-5 баллов);
- Удобство в использовании и дизайн (0-5 баллов).

Помимо общих критериев учитываются особенности той номинации, в которой представлена работа:

1. Программирование: мобильные приложения любой тематики, отличающиеся функциональной продуманностью и сложностью технической реализации.

2. Игровое приложение: мобильные игровые приложения с интересной идеей и логикой игры, проработанным дизайном.

3. Социальное приложение: приложения, помогающие решить важную социальную проблему общества.

4. Каждая работа может быть представлена только в одной номинации.

2.5. Методические обеспечение программы

Приемы и методы организации занятий.

Методы организации и осуществления занятий

1. Перцептивный акцент:

- а) словесные методы (*рассказ, беседа, инструктаж*);
- б) наглядные методы (*демонстрации мультимедийных презентаций*);
- в) практические методы (*упражнения, задачи*).

2. Гностический аспект:
- а) иллюстративно- объяснительные методы;
 - б) репродуктивные методы;
 - в) проблемные методы (методы проблемного изложения) дается часть готового знания;
 - г) эвристические (частично-поисковые) с возможностью выбора вариантов;
 - д) исследовательские – учащиеся сами открывают и исследуют знания.
3. Логический аспект:
- а) индуктивные методы, дедуктивные методы, продуктивный;
 - б) конкретные и абстрактные методы, синтез и анализ, сравнение, обобщение, абстрагирование, классификация, систематизация, т.е. методы как мыслительные операции.
4. Управленческий аспект:
- а) методы учебной работы под руководством учителя;
 - б) методы самостоятельной учебной работы учащихся.

Методы стимулирования и мотивации деятельности

1. Методы стимулирования мотива интереса к занятиям геймификация образовательного процесса, сюжетная игровая составляющая курса, познавательные задачи, учебные дискуссии.

2. Методы стимулирования мотивов долга, сознательности, ответственности, настойчивости: убеждение, требование, приучение, упражнение, поощрение.

2.6. Календарный график воспитательной работы

№ п/п	Название мероприятия/события	Форма проведения	Сроки проведения
1.	День окончания Второй мировой войны. «Конец войны, начала мира».	Инфочас	3 сентября
2.	День солидарности в борьбе с терроризмом.	Инфочас	3 сентября
3.	Проведение организационного родительского собрания по объединениям по теме «Взаимосвязь дополнительного образования и профессионального самоопределения»	Собрание	11-17 сентября
4.	Посещение предприятия/музея или встреча с представителями организаций (по мере договорённости)	Экскурсия	В течение месяца
5.	Научные и технические достижения, открытия, памятные даты	Публикация интересных фактов в пабликах социальных сетей	В течение месяца

6.	Международный день пожилых людей	Изготовление анимационных и графических открыток	1-2 октября
7.	Всемирный день защиты животных. «Мы в ответе за тех, кого приручили».	Тематическая викторина онлайн/ очно.	4 октября
8.	Международный день учителя «Я творчество своё дарю».	Тематические занятия по изготовлению поздравлений.	5 октября
9.	День отца. «Делай вместе с папой»	Краевой выходной	15 октября
10.	Посещение предприятия/музея или встреча с представителями организаций (по мере договорённости)	Экскурсия	В течение месяца
11.	Научные и технические достижения, открытия, памятные даты	Публикация интересных фактов в пабликах социальных сетей	В течение месяца
12.	День народного единства.	Публикация в социальных сетях.	3 ноября
13.	День матери в России. «Подарок маме».	Занятие в объединениях.	26 ноября
14.	День Государственного герба Российской Федерации. «История герба России»	Тематическое занятие/викторина	30 ноября
15.	Посещение предприятия/музея или встреча с представителями организаций (по мере договорённости)	Экскурсия	В течение месяца
16.	Научные и технические достижения, открытия, памятные даты	Публикация интересных фактов в пабликах социальных сетей	В течение месяца
17.	День неизвестного солдата.	Публикация в социальных сетях.	1 декабря
18.	День Героев Отечества.	Публикация в социальных сетях.	9 декабря
19.	День Конституции Российской Федерации.	Публикация в социальных сетях.	12 декабря
20.	Конкурс поделок и открыток к Новому году среди учащихся ЦТЦО «ТЕХНО-IT-куб»	Конкурс	декабрь
21.	Посещение предприятия/музея или встреча с представителями организаций (по мере договорённости)	Экскурсия	В течение месяца

22.	Научные и технические достижения, открытия, памятные даты	Публикация интересных фактов в пабликах социальных сетей	В течение месяца
23.	80 лет со Дня полного освобождения Ленинграда от фашистской блокады (1944 год) «Дорога к жизни»	инфочасы в объединениях	26-27 января
24.	Посещение предприятия/музея или встреча с представителями организаций (по мере договорённости)	Экскурсия	В течение месяца
25.	Научные и технические достижения, открытия, памятные даты	Публикация интересных фактов в пабликах социальных сетей	В течение месяца
26.	День разгрома советскими войсками немецко-фашистских войск в Сталинградской битве.	Инфочас/викторина	2 февраля
27.	День российской науки, 300-летие со времени основания Российской Академии наук (1724 г).	Тематическая викторина (онлайн/очно).	8 февраля
28.	День защитника Отечества, занятия в объединениях/выставка работ учащихся.	Публикация в соцсетях.	21-24 февраля
29.	Посещение предприятия/музея или встреча с представителями организаций (по мере договорённости)	Экскурсия	В течение месяца
30.	Научные и технические достижения, открытия, памятные даты	Публикация интересных фактов в пабликах социальных сетей	В течение месяца
31.	Международный женский день, занятия в объединениях/выставка работ учащихся.	Публикация в соцсетях.	6-7 марта
32.	Посещение предприятия/музея или встреча с представителями организаций (по мере договорённости)	Экскурсия	В течение месяца
33.	Научные и технические достижения, открытия, памятные даты	Публикация интересных фактов в пабликах социальных сетей	В течение месяца
34.	«К звездам!»	инфочасы в объединениях	11-12 апреля
35.	«Космос – это мы»	викторина (онлайн в пабликах/очная)	11-12 апреля
36.	День космонавтики	Информационные сообщения в пабликах	11-12 апреля

		социальных сетей	
37.	Всемирный день здоровья.	Публикация в соцсетях	7 апреля
38.	Посещение предприятия/музея или встреча с представителями организаций (по мере договорённости)	Экскурсия	В течение месяца
39.	Научные и технические достижения, открытия, памятные даты	Публикация интересных фактов в пабликах социальных сетей	В течение месяца
40.	«Вам, ветеранам»	подготовка поздравительных работ учащимися	2-9 мая
41.	День победы	Занятия в объединениях/ инфочасы/викторина Информационные сообщения и поздравления в пабликах социальных сетей	6-8 мая
42.	Посещение предприятия/музея или встреча с представителями организаций (по мере договорённости)	Экскурсия	В течение месяца
43.	Научные и технические достижения, открытия, памятные даты	Публикация интересных фактов в пабликах социальных сетей	В течение месяца
44.	Международный день защиты детей	Информационное сообщение в пабликах соцсетей	1 июня
45.	День России (12 июня)	Информационное сообщение, поздравление в пабликах соцсетей	11-12 июня
46.	Всемирный день охраны окружающей среды (День эколога, 5 июня)	Информационное сообщение в пабликах соцсетей	5 июня
47.	День памяти и скорби – день начала Великой Отечественной войны Инфочасы	Информационное сообщение в пабликах соцсетей	22 июня
48.	Научные и технические достижения, открытия, памятные даты	Публикация интересных фактов в пабликах социальных сетей	В течение месяца

Список литературы

1. Голощапов А. Л., Google Android. Создание приложений для смартфонов и планшетных ПК. – 2-е изд., перераб. и доп. – СПб.: БХВ-Петербург, 2014. – 928 с.
2. Гриффитс, Д., Гриффитс, Дж., Head First. Программирование для Android. 2-е изд. – СПб.: Питер, 2018. – 912 с.;
3. Дейтел П., Дейтел Х., Уолд А., Android для разработчиков. 3-е изд. — СПб.: Питер, 2016. — 512 с.;
4. Лафоре Р., Структуры данных и алгоритмы в Java. Классика Computers Science. 2-е изд. — СПб.: Питер, 2013. — 704 с.
5. Седжвик Р., Уэйн К., Алгоритмы на Java 4-е изд. – Перевод с англ. А.А. Моргунова. – М.: Вильямс, 2013. – 848 с.;
6. Сьерра, К., Бейтс, Б., Изучаем Java 2 изд. – М.: Эксмо, 2012 – 708 с.;
7. Эккель Б., Философия Java. 4-е полное изд. – СПб.: Питер, 2015. – 1168 с.

Приложение №1

Примерные задания для тестирования

Модуль 1.

Вопрос 1; Балл: 1,00

Текст вопроса: "Соберите" перетаскиванием пустую функцию main

- main
- String[]
- (
- static
- public
-)
- params
- {}
- void

Вопрос 2; Балл: 1,00

Текст вопроса: Переменной типа double можно присваивать значения...

Выберите один или несколько ответов:

- 56.7f
- 56.7
- -567
- "56.7"
- 567

Вопрос 3; Балл: 1,00

Текст вопроса: Вызов, осуществляющий вывод на экран значений параметров a и b, каждое с новой строки, записывается ...

- out.
-)

- "\n"+
- (
- print
- a+
- +b
- System.
- "\n"

Вопрос 4; Балл: 1,00

Текст вопроса: Какое минимальное целое число вместо подчеркивания нужно вписать, чтобы переменная a после цикла приняла значение 256?

- int a = 2;
- while (a < ____){
- a *= a;
- }

Введите числовой ответ или символы

TL - если цикл зависнет в любом случае;

ER - если программа при каких-то значениях закончит работу корректно, но выполнить условие невозможно.

Вопрос 5; Балл: 1,00

Текст вопроса: Напишите условие на языке Java, которое соответствует утверждению. «Точка с координатами (x; y) заключена между параболой $y = x^2 - 4$ и прямой $y = x$ »

Только условие (логическое выражение).

Например, корректным по форме, но не по содержанию ответом является $x == y \ \&\& \ (x * x < 10 \ || \ x != y)$

Границы включаются.

Вопрос 6; Балл: 1,00

Текст вопроса: Аналогом фрагмента кода

```
if (a == 0 || b == -1) {
    f();
}
```

Является:

Выберите один ответ:

```
if (b == -1) {
    f();
} else if (a == 0) {
    f();
}
```

```
if (a == 0 && b == -1) {
    f();
}
```

```
if (a != 0 && b != -1) {
    f();
}
```

```
}
```

```
if (a != 0) {  
  } else if (b != -1) {  
  f();  
}
```

```
if (a == 0)  
  if (b == -1) {  
  f();  
}
```

Вопрос 7; Балл: 1,00

Текст вопроса Какое значение НЕ может быть получено в результате выполнения фрагмента программы

```
switch (i) {  
  case 0: System.out.print(i);  
  case 3: System.out.print(i);  
  case 5: System.out.print(i);  
  default: System.out.print("FAIL");  
}
```

Выберите один или несколько ответов:

FAIL

5FAIL

3

33FAIL

000FAIL

035FAIL

5

Вопрос 8; Балл: 1,00

Текст вопроса: Какое ЦЕЛОЕ значение необходимо указать на месте пропуска для того, чтобы этот фрагмент вывел 500 строк.

```
int i = 1;  
while (i < ) {  
  System.out.println(i);  
  i++;  
}
```

Вопрос 9; Балл: 1,00

Текст вопроса: "Соберите" перетаскиванием строку, которая вычисляет факториал числа b (1*2*3*...*b) в переменной f.

(Все переменные объявлены заранее.)

-)
- --b
- f = 1;
- (
- for

- b > 1
- f*=b;
- }
- ;
- {

Вопрос 10; Балл: 1,00

Текст вопроса: Выберите конструкции, идентичные по вычислению значения переменной m фрагменту:

```
int m = 1;
for(int c = 1; c < 12; ++c) {
m = m*c;
}
```

Выберите один или несколько ответов:

```
int m = 1, c = 1;
while (c < 100){
m *=c;
if (c == 11) continue;
c++;
}
```

```
int c = 1, m = 1;
do {
c++;
m *= c;
if (c!=11)continue;
break;
} while (c >0);
```

```
int m = 1, c = 1;
while (c < 100){
m *=c;
if (c == 12) break;
}
```

```
int m = 1,
c = 1;
while (c < 100){
m *=c;
if (c == 11) break;
c++;
}
```

Вопрос 11; Балл: 1,00

Текст вопроса: Укажите варианты корректного по синтаксису объявления массива a. Выберите один или несколько ответов:

```
double a;
```

```
float a{ };  
int [] a, b;  
int [] a;
```

Вопрос 12; Балл: 1,00

Текст вопроса: Укажите все варианты, которые можно указать на месте пропуска для успешной компиляции кода.

```
float array[] = new float[10];
```

```
for (_____ i: array)  
    System.out.print(i + " ");
```

Выберите один или несколько ответов:

```
float[]  
float  
long  
float{ }  
double
```

Вопрос 13; Балл: 1,00

Текст вопроса: Выберите верные по синтаксису определения функций.

Выберите один или несколько ответов:

```
void r(){  
    int i = 5;  
    return i;  
}
```

```
int r(int s){  
    float i = 5f;  
    return (int)i;  
}
```

```
double r(int s, boolean w){  
    double i = 5.0;  
    return s;  
}
```

```
boolean r(){  
    int i = 5;  
}
```

```
int r(float v){  
    int i = 5;  
    return i;  
}
```

```
void r(int s){  
    int i = 5;  
}
```

Вопрос 14; Балл: 1,00

Текст вопроса: Какое целое число можно поставить вместо пропуска, чтобы фрагмент вывел 0?

```
public class Test {
    static int a=0;
    public static void main(String[] args) {
        int max = __;
        for(int a = 1; a < max;){
            a += a;
        }
        System.out.println(a);
    }
}
```

Выберите один ответ:

Только 1000

Только 17

Среди ответов нет правильного

Любое

Вопрос 15; Балл: 5,00

Текст вопроса

- $j < 0; j++$
- $i < n$
- $i < j$
- $j \geq 0$
- m
- $m - 1$
- 2
- 1
- 0
- $i + j == 0$
- $i \% 2 == 0$

"Допишите" перетаскиванием код так, чтобы он заполнял массив "змейкой"

Нужно использовать не все элементы

```
int n = in.nextInt(), m = in.nextInt(), k = 0;
```

```
int[][] matrix = new int[n][m];
```

```
for (int i = ; ; i++) {
    if ( ) {
        for (int j = 0; j < m; j++) {
            matrix[i][j] = k;
            k++;
        }
    } else {
        for (int j = ; ; j--) {
            matrix[i][j] = k;
            k++;
        }
    }
}
```

```

        }
    }
}

```

То есть, например, при вводе 5 4 получалось

```

0 1 2 3
7 6 5 4
8 9 10 11
15 14 13 12
16 17 18 19

```

Вопрос 16; Балл: 1,00

Текст вопроса: Напишите программу, которая выполняет "сжатие массива целых чисел" – переставляет все нулевые элементы в конец.

При этом все ненулевые элементы располагаются в начале в том же порядке.

Возможно решать задачу по-разному, например, использовать дополнительные массивы или даже не использовать массивы совсем.

Главное - вывести получившуюся последовательность.

Вы должны написать полностью программу на Java, один public класс с именем Main в пакете по умолчанию (в коде отсутствует подстрока package) с функцией main(),

которая решает задачу.

Ввод осуществляется с консоли, вывод - на консоль.

Программа должна работать точно по протоколу, который показан в примерах.

Например, нельзя выводить ничего лишнего, скажем, фразу "Введите N".

Входные данные

Первая строка содержит размер массива N.

Во второй строке через пробел задаются N целых чисел – элементы массива.

Гарантируется, что входные данные корректны и $N < 10001$.

Выходные данные

Программа должна вывести в одну строчку все элементы получившегося массива, разделив их пробелами.

For example:

Input Result

```

6
1 0 2 0 3 4
      1 2 3 4 0 0

```

```

6
1 2 3 4 5 6
      1 2 3 4 5 6

```

Модуль 2

Вопрос 1; Балл: 4,00

Текст вопроса: В результате работы программы

```

import                                     java.util.Scanner;
public                                     class                               Translite                               {

```

```

public static void main(String[] args) {
    System.out.println(Integer.toBinaryString((new
Scanner(System.in)).nextInt()));
}
}

```

было выведено **10111**.

На вход программе было подано значение...

Вопрос 2; Балл: 4,00

Текст вопроса: Для строк

```
String s1 = new String("My IT school");
String s2 = Ответ_____;
```

выражение **s1 == s2** имеет значение **true**.

Вставьте пропущенный фрагмент кода.

(Укажите максимально краткий вариант)

Вопрос 3; Балл: 4,00

Текст вопроса: Расположите методы жизненного цикла активности, в порядке от создания до удаления.

- onStart()
- onDestroy()
- onResume()
- onStop()

Вопрос 4; Балл: 5,00

Текст вопроса: Какие модификаторы доступа существуют в Java?

Выберите один или несколько ответов:

- forall
- hidden
- private
- default (без спецификатора)
- public
- protected

Вопрос 5; Балл: 5,00

Текст вопроса:

```

class Base{
    void f(){/*...*/}
}
class Derived extends Base{
    _____ void f(){/*...*/}
}

```

Какие модификаторы **могут быть** применены для метода **f()** в классе **Derived**?

Выберите один или несколько ответов:

- public
- default (без модификатора)
- private

- protected

Вопрос 6; Балл: 5,00

Текст вопроса: Соберите "перетаскиванием" конструктор без аргументов класса **MyClass**.

- }
- a
- MyClass
- = -1;
- this.
- public
- (
- {
-)

Вопрос 7; Балл: 5,00

Текст вопроса

В классе **Amount** (счет) банка "Счастливым" открывают счета сразу с суммой 7 рублей 77 копеек. Создайте счет в банке - объект **amount1** класса **Amount** при помощи конструктора без аргументов.

Так чтобы, например, фрагмент кода

```
//YOUR CODE will be placed HERE
```

```
//...
```

```
amount1.print();
```

```
вывел "7 rub 77 kop"
```

(Вам нужно только *создать объект amount1 конструктором без аргументов*, сам класс, его конструктор, метод print() писать **не нужно**, считается, что они работают корректно.)

Вопрос 8; Балл: 5,00

Текст вопроса: Выберите верные утверждения

Выберите один или несколько ответов:

- Наследование это один из ключевых принципов ООП
- Класс от которого происходит наследование называется суперкласс
- Интерфейсы нельзя наследовать от других интерфейсов
- В java допускается до трех суперклассов у подкласса.
- Для указания наследования в коде используется ключевое слово **extends**

- Ключевое слово **super** используется для вызова конструктора суперкласса из любого метода подкласса.

Вопрос 9; Балл: 5,00

Текст вопроса: Имеется класс Person

```
class Person{
    public String name;
    public int age;
}
```

Опишите его подкласс **Student** таким образом, чтобы в объекте этого типа были доступны следующие публичные элементы:

- строковая переменные **name**(имя) и целочисленные переменные **age**(возраст), **course** (номер курса),
- методы **expelled** (отчислить) и **enrolled** (зачислить), оба без параметров и возвращаемого значения.

Примечание. Используются типы данных `int` и `String`. Описываемые методы могут иметь самые простые или даже пустые тела. Например: `void func(){}` или `int func(){return 0;}`

Вопрос 10

Пока нет ответа

Балл: 5,00

Текст вопроса: Какое ключевое слово необходимо указать на месте пропуска для запрета наследования метода **next()**?

```
public Ответ  void next(int i){
    /*...*/
}
```

Вопрос 11; Балл: 5,00

Текст вопроса: Создайте класс **Cat**, наследник **Animal**, и реализуйте в нем полиморфный метод **speak()**, возвращающий строку **"Meow"**.

```
abstract class Animal{
    public abstract String speak();
}
```

Чтобы, допустим, код

```
Animal cat = new Cat();
```

```
System.out.println(cat.speak());
```

вывел **Meow**

Вопрос 12; Балл: 5,00

Текст вопроса: Имеется интерфейс

```
interface Runnable{
    public void run();
}
```

Опишите класс **MyClass**, имплементирующий этот интерфейс и также имеющий:

- публичное поле логического типа - **running**,
- два публичных метода без возвращаемого значения и параметров - **start** и **stop**.

Примечание. Используется типы данных `boolean`. Объявляемый интерфейс должен иметь минимально возможное количество полей и методов.

Вопрос 13; Балл: 5,00

Текст вопроса: Поставьте в соответствие атрибуту объекта его описание

android:hint

android:text

android:layout_height
android:layout_width
android:layout_gravity

Вопрос 14; Балл: 5,00

Текст вопроса: Для объектов:

EditText

input;

TextView output;

соберите команду, которая считывает текст из поля ввода и выводит его в текстовое поле.

output .

input . .);

Вопрос 15; Балл: 5,00

Текст вопроса: Добавьте в класс **MyClass**

```
//Fix the class
class MyClass{
    void f(){
        count++;
        System.out.print(count + " ");
    }
}
```

объявление переменной **count** так, чтобы в ней подсчитывалось количество вызовов функции **f()**.

Чтобы, например, фрагмент кода

```
MyClass obj1 = new MyClass(), obj2 = new MyClass();
obj1.f();
obj2.f();
obj1.f();
```

вывел **1 2 3**.

В ответе приведите **исправленный класс целиком**.

Вопрос 16; Балл: 8,00

По разметке и файлу **string.xml** напишите класс Активности, чтобы при тапе по кнопке выводился тост со строкой **text** из строковых ресурсов.



Файл

res/layout/activity_main.xml

```
<!-- ?xml version="1.0" encoding="utf-8"? -->
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <Button
        android:id="@+id/button"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
```

```

        android:text="@string/click_me"
    </LinearLayout>

```

Файл **res/values/string.xml**

```

<resources>
    <string name="app_name">Test</string>
    <string name="click_me">Click me</string>
    <string
        name="text">Hello!</string>
</resources>

```

Приведите в решении только класс Активности с именем **MainActivity**.

```

package test.itschool.samsung.ru.test;
import android.app.Activity;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.Toast;
//YOUR CODE will be placed HERE
//....

```

Верным по форме, но не по содержанию ответом является код

```

public class MainActivity extends Activity{
}

```

Модуль 3

Вопрос 1; Балл: 4,00

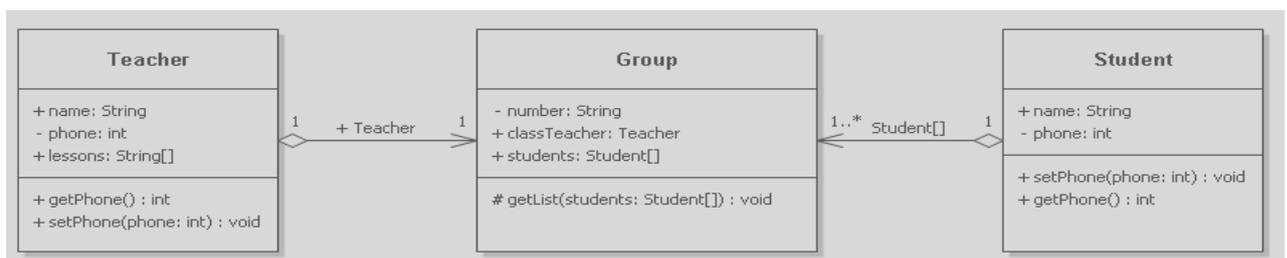
Текст вопроса: Выберите верные утверждения о процессе объектного проектирования.

Выберите один или несколько ответов:

- UML - это универсальный язык моделирования
- UML не может использоваться для описания иерархий классов
- Проектирование — необязательный этап разработки ООП приложений
- Существует отношение классов агрегирование

Вопрос 2; Балл: 5,00

Текст вопроса: Постройте описание класса Group по UML диаграмме. Объявляемый класс должен иметь минимальное количество полей и методов.



Примечание. Используйте тип данных String, Teacher и Student.

Описываемые методы могут иметь самые простые или даже пустые тела.

Например: void f() {} или int f() {return 0;}

Вопрос 3; Балл: 4,00

Текст вопроса: Имеется код. Выберите верные утверждения.

```
int number;
int zero;
try {
    number = 1;
    zero = 0;
    int result = number / zero;
    System.out.println("Message1");
} catch (ArithmeticException e) {
    System.out.println("Message2");
}
System.out.println("Message3");
```

Выберите один или несколько ответов:

- Сообщение Message2 не будет показано.
- Сообщение Message3 не будет показано.
- Этот код содержит критический блок.
- Сообщение Message1 не будет показано.
- При выполнении этого кода не будет создаваться объект класса наследника Exception
- В этом коде присутствует блок проверки.

Вопрос 4; Балл: 5,00

Текст вопроса

- catch
- try
- throwable
- finally
- throw
- final
- Exception

Заполните поля для правильной обработки исключений

```
PrintWriter writer = null;
{
    writer = new PrintWriter(new FileWriter("filename.txt"));
    writer.write("Something");
}  (IOException e) {
    System.out.println("File operation failed: "+e.getMessage());
}  {
    try {writer.close();} catch (Exception e) {/*ignore*/}
}
```

Вопрос 5; Балл: 5,00

Текст вопроса: В `Logcat` вывелось сообщение

12-28 20:25:12.272 32347-32347/? D>Loading: Images_loaded

Напишите вызов метода класса **Log**, которую написал программист для мониторинга загрузки изображений.

Вопрос 6; Балл: 5,00

Текст вопроса: Обычно при отладке предотвращают ошибки.

В этом задании, наоборот, вам надо **сгенерировать ошибку**

Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: index=3

то есть **написать произвольный фрагмент кода, который приведет к такой ошибке работы программы** (при тестировании ваш код будет размещен в функции класса и эта функция вызвана)

Вопрос 7; Балл: 4,00

Текст вопроса: Предположим, в нашей программе создается экземпляр класса `File`:

```
String pathToFile = ...;
```

```
File file = new File(pathToFile);
```

Что при этом будет создано на диске?

Выберите один ответ:

- Файл или директория, в зависимости от наличия символа "/" в конце переданной строки `pathToFile`.
- Файл
- Ничего не создается
- Директория

Вопрос 8; Балл: 5,00

Текст вопроса: Опишите функцию по приведенному ниже образцу, которая выведет информацию о файле с именем в первом аргументе. Требуется вывести на консоль размер и абсолютный путь к файлу.

```
public void getFileInfo(String fileName){
```

```
// necessary code
```

```
}
```

текст функции целиком запишите в поле результата

Вопрос 9; Балл: 5,00

Текст вопроса: Заполните недостающие фрагменты для создания объекта `vesti` - экземпляра класса `Paper`

```
public class InfoService {
```

```
    class Paper {
```

```
        public String title;
```

```
        public int sheets;
```

```
    }
```

```
/* ...*/
```

```
}
```

```
InfoService tvChannel = new _____;
```

InfoService.Paper vesti = ____ . new ____ ;

Вопрос 10; Балл: 5,00

Текст вопроса: Реализуйте локальный внутренний класс NumberPhone, в блоке метода main, для работы с телефонными номерами.

Класс NumberPhone должен иметь конструктор, с принимаемым параметром String phone и метод validate, с возвращаемым значением типа boolean, где true - номер верный, false - нет.

Номер принято считать верным если:

- номер начинается с +7;
- в номере 10 цифр без добавочного (без +7).

Например, номер +79990001111 является правильным, а +7999555 или 89990001111 - нет.

```
import java.util.Scanner;
```

```
public class Main {
```

```
    public static void main(String[] args) {  
        //YOUR CODE will be placed HERE  
        //...
```

```
        Scanner in = new Scanner(System.in);
```

```
        NumberPhone numberPhone = new NumberPhone(in.nextLine());
```

```
        System.out.print(numberPhone.validate());
```

```
    }
```

```
}
```

В качестве ответа необходимо указать полный код класса NumberPhone.

Правильным по форме, но не по содержанию будет являться ответ вида

```
class NumberPhone {
```

```
}
```

Вопрос 11; Балл: 5,00

Текст вопроса

- makeTextClick()
- getApplicationContext()
- View.OnClickListener()
- this.setOnClickListener()
- onClickClickListener()

Соберите обработчик события кнопки, при нажатии на которую появляется "длинный" тост с надписью "Hello!".

```
Button b = (Button) findViewById(R.id.button);
```

```
    b.____) new ____ ._____{
```

```
        public void ____ (View v) {
```

```
            Toast.____ (____, "Hello!", Toast.LENGTH_LONG).____;
```

```
        }
```

```
    });
```

Вопрос 12; Балл: 5,00

Текст вопроса: Многопоточные Android-приложения можно реализовать с использованием наследника класса

Выберите один или несколько ответов:

- MultiTask
- Parallel
- AsyncTask
- Thread
- AnotherProcess

Вопрос 13; Балл: 5,00

Текст вопроса: В целях монетизации при старте приложения должна показываться реклама.

В вашем классе есть функция **showAd()**.

Она не выбрасывает исключений, которые необходимо перехватывать.

Естественно, ее нужно запускать в новом потоке, чтобы пользователь смог отменить показ рекламы.

```
void startApplication(){
    //YOUR CODE will be placed HERE
    //...
}
```

Напишите строки для запуска функции **showAd()** в новом потоке.

Вопрос 14; Балл: 5,00

Текст вопроса: Подберите каждому методу класса Paint описание его функционала

- | | |
|-------------------|---------|
| setTextSize() | Ответ 1 |
| setColor() | Ответ 2 |
| setStyle() | Ответ 3 |
| setSubpixelText() | Ответ 4 |
| setAntiAlias() | Ответ 5 |

Вопрос 15; Балл: 5,00

Текст вопроса

- getWidth()
- drawLine
- setStyle
- 30
- setColor
- paint
- drawCircle
- 100
- getHeight()

Допишите недостающие фрагменты в методе onDraw() так, чтобы по диагонали экрана были нарисованы 20 разноцветных не пересекающихся кругов радиуса 30 пт с центром на диагонали и изменением цвета на 100:

```
public void onDraw(Canvas c) {
    super.onDraw(c);
    Paint paint = new Paint();
```

```

paint._____ (Paint.Style.FILL);
int color=Color.RED;
int x = _____/20;
int y = getHeight()/20;
for (int i= 1; i<21;i++){ color+=100;
paint._____ (color);
c._____ (i*x,i*y,_____,paint);
}

```

Вопрос 16; Балл: 8,00

Текст вопроса:



Приложение SurfaceViewTest состоит из активности, на которую устанавливается главным View объект класса **TestSurfaceView**, подкласса **SurfaceView**

```

public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(new TestSurfaceView(this));
    }
}

```

Ваша задача написать сам класс **TestSurfaceView** таким образом, чтобы фон стал синим и при тапе по экрану начал появляться желтый круг с центром в точке нажатия, увеличивая свой радиус с нуля по 5 пикселей через каждую секунду. При нажатии в другом месте круг должен начать появляться заново (на экране должен "расти" только один круг, в центре последнего нажатия). При организации потоков используйте для задержки только метод **sleep()** класса **Thread**. **Обязательно используйте этот метод при организации потоков хотя бы с минимальной задержкой.** Для установки цвета используйте только константы класса **Color**. Для определения размера canvas используйте методы **getWidth()** и **getHeight()** класса **Canvas**, а не **View** (**SurfaceView**). Подробнее см. на вкладке "Разрешенные методы" наверху страницы. Для проверки вашей программы всем техническим требованиям можно сдать код в специальный тест (откроется в новом окне).

Приведите в решении только класс с именем **TestSurfaceView** и вспомогательные классы, если решение состоит из нескольких классов.

В качестве заготовки для ответа можно использовать этот шаблон

```

class TestSurfaceView extends SurfaceView implements SurfaceHolder.Callback {
public TestSurfaceView(Context context) {
    super(context);
    getHolder().addCallback(this);
}
}

```

```

}
    @Override
    public void surfaceChanged(SurfaceHolder holder, int format, int width,
int height) {
}
    @Override
    public void surfaceCreated(SurfaceHolder holder) {
}
    @Override
    public void surfaceDestroyed(SurfaceHolder holder) {
}
}
}

```

Модуль 4

Вопрос 1; Балл: 1,00

Текст вопроса: Ниже приведено объявление массива. Выберите верные утверждения.

```

Integer [] array;
array =new Integer[10];
for(int i=0;i<array.length;i++)
    array[i]=(new Random()).nextInt();

```

Выберите один или несколько ответов:

- В ячейки указанного массива можно присваивать значения любого типа
- Значения в приведенном массиве можно отсортировать при помощи функции sort класса Arrays
- Методы класса Collections можно использовать для операций с приведенным массивом
- В приведенном массиве нельзя добавить дополнительные ячейки

Вопрос 2; Балл: 2,00

Текст вопроса

- next()
- String
- Comparator
- Iterator
- hasNext()
- remove()
- nextStudent()

В приведенном ниже коде массив студентов создается, и выводится на экран. Заполните поля для правильной работы кода.

```

ArrayList<Student> array=new ArrayList();
for(int i=0;i<10;i++){
    String name=new Scanner(System.in).next();
    array.add(new Student(name));
}

```

```

_____<Student> iterator=array.iterator();
while(iterator._____) {
    Student student=iterator._____$
    System.out.println(student);
}

```

Вопрос 3; Балл: 1,00

Текст вопроса: В двусвязном линейном списке, состоящим из 10 элементов, необходимо поменять **первый** и **последний** элементы местами. Количество ссылок, которое будет изменено

Вопрос 4; Балл: 1,00

Текст вопроса: Выберите верные утверждения
Выберите один или несколько ответов:

- Класс LinkedList реализует двунаправленную очередь
- Класс LinkedList реализует дисциплину обслуживания FIFO
- Для LinkedList сложность операции add(E) - O(n)
- Отсортировать список LinkedList можно при помощи метода sort

класса Arrays

Вопрос 5; Балл: 2,00

Текст вопроса

- this
- dayArr
- AppCompatActivity
- Activity
- Adapter
- String
- dayAdapter

Заполните пропуски так, что в результате работы представленного кода на ListActivity с именем DayList появится список дней недели из массива dayArr.

```

public class DayList extends ListActivity {
    String[] dayArr = {"пн", "вт", "ср", "чт", "пт", "сб", "вс"};
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        ArrayAdapter<____> dayAdapter = new ArrayAdapter<>(____,
android.R.layout.simple_list_item_1, ____);
        setListAdapter(____);
    }
}

```

Вопрос 6; Балл: 1,00

Текст вопроса: Программа получает на вход последовательность целых чисел и строит из них двоичное дерево поиска. **Балансировка дерева не производится.**

В ответе укажите единственное число - высоту получившегося дерева для следующей последовательности чисел, считая, что дерево из одного элемента имеет высоту 1:

2 3 5 6 8

Вопрос 7; Балл: 2,00

Текст вопроса: Создайте дерево (TreeSet) с именем **tree**, для объектов типа User, отсортированных по **возрастанию** значения поля **name**.

```
import java.util.Comparator;
import java.util.TreeSet;
class User {
    String name;
    int age;
    User(String n, int a) {
        name = n;
        age = a;
    }
}
class Main {
    public static void main(String[] args) {
        //your code...
    }
}
```

Вопрос 8; Балл: 1,00

Текст вопроса: Расположите в верном порядке тело рекурсивной функции, вычисляющей произведение элементов списочного массива

- return d*multAList(x); }
- int d = x.remove(0);
- if (x.isEmpty())
- static long multAList(ArrayList x) {
- return 1;
- }
- else {

Вопрос 9; Балл: 2,00

Текст вопроса: Инициализируйте объект *a* таким образом, чтобы в результате работы программа

```
import java.util.Collections;
import java.util.LinkedList;
public class Test {
    public static void main(String[] args) {
        LinkedList<String> a = new LinkedList<>();
        a.add("spring");
        // write your code here
        System.out.println(Collections.binarySearch(a, "spring"));
    }
}
```

выдала значение 3.

Вопрос 10; Балл: 1,00

Текст вопроса: Расположите элементы массива после сортировки

```
Integer [] a = { 16, 28, 60, 22, 67, 54, 11 };
```

```
Arrays.sort(a, a.length/2, a.length);
```

- 11
- 54
- 22
- 28
- 60
- 67
- 16

Вопрос 11; Балл: 1,00

Текст вопроса: Вывод коллекции **collection** осуществляется так

```
for (Object obj: collection){
```

```
    System.out.print(obj + " ");
```

```
}
```

Каждому заполнению коллекции подберите вывод данных коллекции этим фрагментом

```
LinkedList collection = new LinkedList<>();
```

```
collection.add(3);
```

```
collection.add(21);
```

```
collection.add(1);
```

```
collection.add(25);
```

```
HashSet collection = new HashSet();
```

```
collection.add("3");
```

```
collection.add("21");
```

```
collection.add("1");
```

```
collection.add("25");
```

```
TreeSet collection = new TreeSet();
```

```
collection.add(3);
```

```
collection.add(21);
```

```
collection.add(1);
```

```
collection.add(25);
```

```
TreeSet collection = new TreeSet();
```

```
collection.add("3");
```

```
collection.add("21");
```

```
collection.add("1");
```

```
collection.add("25");
```

Вопрос 12; Балл: 2,00

Текст вопроса: В ветеринарной клинике данные о животных хранятся в ассоциативном массиве **pets**, созданном на основе контейнера

TreeMap. Ключом является кличка питомца, значением - имя хозяина.

```
TreeMap<String, String> pets = new TreeMap<>();
```

Каждый день ветеринар обходит своих клиентов в алфавитном порядке. Например по понедельникам он успевает обойти питомцев, клички которых начинаются с латинской буквы "А" и заканчиваются латинской "Н" включительно (**регистр имеет значение**).

Необходимо создать ассоциативный массив **mondayPets** таким образом, чтобы вывод

```
System.out.println(mondayPets);
```

показал ветеринару порядок обхода питомцев в понедельник.

Новый ассоциативный массив должен быть создан на основе выборки необходимых значений из существующего.

Правильным по форме, но не по содержанию является ответ вида

```
TreeMap <String, String> mondayPets = new TreeMap<>();
```

Вопрос 13; Балл: 1,00

Текст вопроса:Для отдельно взятой таблицы в реляционной модели баз данных верно утверждение ...

Выберите один ответ:

- Все строки имеют одинаковое количество столбцов и в каждом столбце хранятся данные одного типа
- Каждая строка содержит произвольное количество однотипных данных
- В одном столбце в разных строках могут быть записаны элементы разного типа
- Каждая строка может иметь собственное количество столбцов

Вопрос 14; Балл: 2,00

Текст вопроса

- INCREMENT
- NULL
- NOT NULL
- KEY
- TEXT
- AUTOINCREMENT
- DEFAULT
- PRIMARY KEY
- INTEGER
- USERS

Таблица users имеет следующие метаданные

name	type	length	is null	auto increment
_id*	INT		false	true
name	TEXT		false	
age	INT		true	false

Дополните запрос CREATE таблицы users в соответствии с представленными метаданными

CREATE TABLE _____(

```
_id INTEGER _____ NOT NULL,  
name _____ NOT NULL,  
age _____);
```

Вопрос 15; Балл: 2,00

Текст вопроса:

from COUNTRIES where STOCK

Соберите перетаскиванием запрос к таблице STOCK со сведениями о компаниях

```
mysql> select * from STOCK;
```

```
+-----+-----+-----+  
| RIC   | COMPANY           | LISTED_ON_EXCHANGE |  
+-----+-----+-----+  
| 6758.T | Sony              | T                   |  
| GOOG.O | Google Inc       | O                   |  
| GS.N   | Goldman Sachs Group Inc | N                   |  
| INFY.BO | InfoSys          | BO                  |  
| VOD.L  | Vodafone Group PLC | L                   |  
+-----+-----+-----+
```

5 rows in set (0.00 sec)

для получения названия компании с RIC равным GOOG.O

```
select COMPANY _____='GOOG.O';
```

Вопрос 16; Балл: 3,00

Текст вопроса:

Вы должны написать полностью программу на Java: необходимые директивы import, один public класс с именем Main в пакете по умолчанию (в коде отсутствует подстрока package) с функцией main(), которая решает задачу. Ввод осуществляется с консоли, вывод - на консоль. Программа должна работать точно по протоколу, который показан в примерах.

Например, нельзя выводить ничего лишнего, скажем, фразу "Введите N".

Дана база данных о продажах некоторого интернет-магазина. Каждая строка входных данных представляет собой запись вида Покупатель товар количество, где Покупатель — имя покупателя (строка без пробелов), товар — название товара (строка без пробелов), количество — количество приобретенных единиц товара.

Создайте список всех покупателей, а для каждого покупателя подсчитайте количество приобретенных им единиц каждого вида товаров.

Входные данные

Вводятся сведения о покупках в указанном формате.

Выходные данные

Выведите список всех покупателей в лексикографическом порядке, после имени каждого покупателя выведите двоеточие, затем выведите список названий всех приобретенных данным покупателем товаров в лексикографическом порядке, после названия каждого товара выведите

количество единиц товара, приобретенных данным покупателем.
Информация о каждом товаре выводится в отдельной строке.
For example:

Input	Result
Ivanov aaa 1	Ivanov:
Petrov aaa 2	aaa 52
Sidorov aaa 3	bbb 3
Ivanov aaa 6	ccc 45
Petrov aaa 7	Petrov:
Sidorov aaa 8	aaa 9
Ivanov bbb 3	bbb 7
Petrov bbb 7	ddd 34
Sidorov aaa 345	Sidorov:
Ivanov ccc 45	aaa 356
Petrov ddd 34	Ziborov:
Ziborov eee 234	eee 234
Ivanov aaa 45	

Модуль 5

Вопрос 1; Балл: 1,00

Текст вопроса: Утилита, которая позволяет узнать активен ли определенный сервер в сети:

Выберите один ответ:

- nslookup
- netstat
- tracert/traceroute
- ping
- ipconfig/ifconfig

Вопрос 2; Балл: 2,00

Текст вопроса

- 113
- 97
- 224
- 0
- 217
- 96
- 170
- 255

По заданным IP-адресу узла и маске определите адрес подсети.

IP-адрес узла: 217.170.113.105

Маска: 255.255.224.0

Вопрос 3; Балл: 1,00

Текст вопроса: Укажите существующие методы HTTP

Выберите один или несколько ответов:

- end
- remove
- post
- reset
- put
- get

Вопрос 4; Балл: 2,00

Текст вопроса: Соберите первую строку ответа Web-сервера перетаскиванием

- /
- 0
- 1
- 200
- HTTP
- OK
- .

Вопрос 5; Балл: 1,00

Текст вопроса: Ниже приведены утверждения о процессе клиент серверного взаимодействия. Выберите верные

Выберите один или несколько ответов:

- Часто взаимодействие между клиентом и сервером происходит при помощи протокола HTTP
- Для клиент-серверного взаимодействия не нужно добавления никаких доп. прав в манифест
- Взаимодействие в клиент-серверной архитектуре происходит по протоколу WiFi
- Для андроид есть простая реализация HTTP клиента от Apache

Вопрос 6; Балл: 2,00

Текст вопроса: В программе используется класс

```
class Staff {
    public String name;
    public int age;
    public String position;
    public BigDecimal salary;
    public List<String> skills;
}
```

Функция **int countSkills()** получает информацию об этом объекте в виде JSON-строки.

Например,

```
{"name":"Anton","age":15,"position":"Junior","salary":100,"skills":["java","xml","python","Android"]}
```

и должна возвращать количество умений сериализованного объекта, например, для данного объекта - **4**.

Допишите функцию
import com.google.gson.*;
//...
int countSkills(String staffJson){
 // YOUR CODE will be paced HERE

Вопрос 7; Балл: 1,00

Текст вопроса

- Gson
- fromJson
- Json
- Serializable
- toJson
- School

Допишите код для сериализации объекта класса School (класс корректно описан ранее и содержит конструктор с двумя параметрами) в формат json с использованием библиотеки com.google.gson

```
School u1 = new _____(14, "СОШ");  
Gson f = new _____();  
String res = f._____(u1);
```

Вопрос 8; Балл: 1,00

Текст вопроса: Формат данных JSON имеет ряд преимуществ по сравнению с XML, отметьте верные утверждения.

Выберите один или несколько ответов:

- JSON является текстовым форматом, который может прочесть и человек, и машина, а XML - нет
- В JSON можно использовать массивы (arrays), в XML - нет
- Для передачи информации по сети используют оба формата
- Формат XML имеет иерархическую структуру, а JSON не поддерживает иерархичность
- В формате JSON предусмотрен контроль целостности данных, а в XML - нет

Вопрос 9; Балл: 1,00

Текст вопроса: Библиотека Retrofit имеет ряд преимуществ. Выберите верные утверждения

Выберите один или несколько ответов:

- Полученные данные могут быть автоматически преобразованы в экземпляры нужных классов (т.е. десериализованы)
- Отправка запросов и получение ответов может происходить как в асинхронном режиме, так и в синхронном режиме.
- Retrofit содержит встроенные конвертеры для форматов JSON и XML
- Retrofit - единственная библиотека, поддерживающая протокол HTTPS (безопасное соединение)

Вопрос 10; Балл: 2,00

Текст вопроса

List<User>

Call

Callback

Call<User>

Response

"http://my.ru/api/get_users"

"api/get_users"

В представленном примере интерфейс **APIService** используется для создания объекта **Retrofit** и содержит единственный метод **getUsers()** для получения всех пользователей с сервера.

Получение списка пользователей в виде **JSON массива** осуществляется путем отправки **HTTP GET** запроса по адресу - **http://my.ru/api/get_users**. Метод **getUsers()** возвращает список пользователей сериализованный в **List** объектов класса **User**.

Заполните пропуски для корректной работы программы.

```
import retrofit.client.Response;
import retrofit.http.GET;
import retrofit.Call;
public interface APIService {
    @GET(_____)
    _____<_____> getUsers();
}
```

Вопрос 11; Балл: 2,00

Текст вопроса: В приведенном ниже примере создан объект библиотечного класса **Retrofit** на основе интерфейса **UserService** с единственным методом - **getUsers()**.

getUsers() возвращает список всех пользователей полученных с сервера сериализованный в **List** объектов класса **User**.

Заполните пропуски для **синхронного** выполнения запроса на получения списка пользователей с сервера.

ВНИМАНИЕ!!! Не используйте пробельные символы при заполнении!

```
Retrofit retrofit = new Retrofit.Builder()
    .baseUrl("http://myitschool.ru/")
    .addConverterFactory(GsonConverterFactory.create())
    .build();
UserService service = retrofit.create(UserService.class);
_____<_____<_____>><list call = _____ _____>();
call._____();
</list
```

Примеры проектов выпускников: <https://myitschool.ru/apps/index.php>.