Сборник практических и лабораторных работ по
междисциплинарному курсу
«МДК.03.04 Разработка мобильных приложений»

Специальность: 09.02.03 Программирование в компьютерных системах

Содержание

1 Общие сведения

Сборник содержит задания для практических работ для студентов, обучающихся специальности 09.02.03 Программирование в компьютерных системах.

Задачник расчитан на 64 часа практических занятий.

Структурно курс делится на две части:

1. изучение языка Kotlin,

2. изучение разработки на платформе Android.

Изучение курса базируется на том, что основы программирования, объектно-ориентированное программирование и современные подходы к его применению изучены в предыдущих дисциплинах. Соответственно, работы предназначены для освоения новых навыков, в частности функционального подхода к программированию.

В практических и лабораторных работах предполагается использование языка Kotlin в средах IntelliJ IDEA и Android Studio. При этом необходимо соблюдать Coding Conventions (требования к стилю кода).

Перед сдачей работы добейтесь, чтобы среда не выдавала предупреждений при запуске подпункта **Inspect code** пункта меню **Analyze**.

Задачи в большей степени расчитаны на освоение возможностей языка, а не на алгоритмические сложности, потому осуществляйте написание кода в соответствии с заданием, а не с целью, чтобы он просто работал.

Обратите внимание на то, что во всех заданиях необходимо проверять корректность входных данных, программа не должна «падать» ни в каких ситуациях.

2 Язык Kotlin

2.1 Практическая работа «Основы языка Kotlin» (2 часа)

Цель первой работы – обеспечить подготовку к выполнению заданий, нацеленных на освоение особенностей языка Kotlin. В первой работе необходимо написать несколько небольших программ, чтобы убедиться в понимании базовых конструкций языка (функция main, ветвления, циклы) и базовые типы (целые, вещественные, логический, символьный, массивы, строки), а также обеспечить успешную настройку среды разработки.

При выполнении заданий обращайте внимание на использование специфических особенностей языка везде, где это возможно: if и when могут быть как операторами, так и частью выражений; фигурные скобки во многих случаях можно опускать; точки с запятой почти никогда не используются, корректно выбирайте, как помечать переменные: ключевым словом var или val.

При выполнении работ обеспечивайте **оптимальность** предлагаемой программы как по скорости, так и по памяти. В случае противоречия между двумя критериями, выбирайте алгоритм, который обеспечивает лучшее быстродействие.

В частности, это обозначает, что нельзя использовать дополнительные строки (в заданиях, кроме как для ввода, строки не нужны), следует избегать сложных структур (списков, множеств). В целом необходимо выполнить задание теми способами, которые использовались в курсе «Основы алгоритмизации и программирования», но на новом языке.

Задание №1

Для данного неотрицательного целого числа (в пределах Int) найдите указанный результат. Осуществите проверку корректности ввода. Оформите программу и устраните все warning.

1. сумма четных цифр
2. сумма нечетных цифр
3. произведение четных цифр
4. произведение нечетных цифр
5. максимальную четную цифру
6. минимальную четную цифру
7. максимальную нечетную цифру
8. минимальную нечетную цифру
9. сумма цифр, кратных трем
10. сумма цифр, некратных трем
11. произведение цифр, кратных трем
12. произведение цифр, некратных трем
13. максимальную цифру, кратную трем
14. минимальную цифру, кратную трем
15. максимальную цифру, некратную трем
16. минимальную цифру, некратную трем

17. сумма цифр, стоящих на четных позициях в числе (если нумеровать цифры с конца): для числа 1234 ответ 4

18. произведение цифр, стоящих на четных позициях в числе (если нумеровать цифры с конца): для числа 1234 ответ 3

19. максимальная цифра среди стоящих на четных позициях в числе (если нумеровать цифры с конца): для числа 1234 ответ 3

20. минимальная цифра среди стоящих на четных позициях в числе (если нумеровать цифры с конца): для числа 1234 ответ 1

21. сумма цифр, стоящих на позициях в числе, номера которых кратны трем (если нумеровать цифры с конца): для числа 1234 ответ 2

22. произведение цифр, стоящих на позициях в числе, номера которых кратны трем (если нумеровать цифры с конца): для числа 1234 ответ 2

23. максимальная цифра среди стоящих на позициях в числе, номера которых кратны трем (если нумеровать цифры с конца): для числа 1234 ответ 2

24. минимальная цифра среди стоящих на позициях в числе, номера которых кратны трем (если нумеровать цифры с конца): для числа 1234 ответ 2

25. максимальная цифра среди стоящих на позициях в числе, номера которых кратны четырем (если нумеровать цифры с конца): для числа 1234 ответ 1

Задание №2

1. С клавиатуры вводится описание массива из 10 элементов в виде:

   номер:*значение*

   однако, порядок указания элементов может быть любой. Выведите все элементы массива в порядке возрастания номеров.

2. В строке указано несколько неотрицательных целых чисел, разделенных пробелами (по одному пробелу между числами). Какие цифры присутствуют в каждом числе?

3. В строке указано несколько неотрицательных целых чисел, разделенных пробелами (по одному пробелу между числами). Какие цифры присутствуют хотя бы в двух числах?

4. В строке указано несколько неотрицательных целых чисел, разделенных пробелами (по одному пробелу между числами). Какие цифры присутствуют ровно в одном числе?

5. В строке указано несколько неотрицательных целых чисел, разделенных пробелами (по одному пробелу между числами). Какие цифры присутствуют ровно в двух числах?

6. В строке указано несколько неотрицательных целых чисел, разделенных пробелами (по одному пробелу между числами). Какие цифры отсутствуют ровно в двух числах?

7. В строке указано несколько неотрицательных целых чисел, разделенных пробелами (по одному пробелу между числами). Какие цифры отсутствуют ровно в одном числе?

8. В строке указано несколько слов, разделенных пробелами (по одному пробелу между словами). Какие символы присутствуют в каждом слове? Предполагается, что все символы в строке имеют код, не больший, чем 127.

9. В строке указано несколько слов, разделенных пробелами (по одному пробелу между словами). Какие символы присутствуют хотя бы в двух словах? Предполагается, что все символы в строке имеют код, не больший, чем 127.

10. В строке указано несколько слов, разделенных пробелами (по одному пробелу между словами). Какие символы присутствуют ровно в одном слове? Предполагается, что все символы в строке имеют код, не больший, чем 127.

11. В строке указано несколько слов, разделенных пробелами (по одному пробелу между словами). Какие символы отсутствуют ровно в одном слове? Предполагается, что все символы в строке имеют код, не больший, чем 127.

12. В строке указано несколько слов, разделенных пробелами (по одному пробелу между словами). Какие символы отсутствуют ровно в двух словах? Предполагается, что все символы в строке имеют код, не больший, чем 127.

13. В строке указано несколько неотрицательных целых чисел, разделенных пробелами (по одному пробелу между числами). Какое количество чисел удовлетворяет условию отсутствия повторяющихся цифр?

14. В строке указано несколько неотрицательных целых чисел, разделенных пробелами (по одному пробелу между числами). Какое количество чисел удовлетворяет условию наличия повторяющихся цифр?

15. В строке указано несколько слов, разделенных пробелами (по одному пробелу между словами). Какое количество слов удовлетворяет условию отсутствия повторяющихся символов? Предполагается, что все символы в строке имеют код, не больший, чем 127.

16. В строке указано несколько слов, разделенных пробелами (по одному пробелу между словами). Какое количество слов удовлетворяет условию наличия повторяющихся символов? Предполагается, что все символы в строке имеют код, не больший, чем 127.

17. В строке указано несколько неотрицательных целых чисел, разделенных пробелами (по одному пробелу между числами). В каком количестве чисел присутствуют все цифры от 0 до 9?

18. Имеется некоторая последовательность цифр от 0 до 9. С клавиатуры вводится 9 строк следующего вида:

    цифра->*цифра*

    Каждая строка обозначает, что после цифры, стоящей до стрелки, в последовательности стоит цифра, стоящая после стрелки.

    Выведите исходную последовательность.

19. В строке указано несколько неотрицательных целых чисел, разделенных пробелами (по одному пробелу между числами). Какие цифры присутствуют в каждом числе дважды?

20. В строке указано несколько неотрицательных целых чисел, разделенных пробелами (по одному пробелу между числами). Какие цифры присутствуют хотя бы в одном числе дважды?

21. В строке указано несколько неотрицательных целых чисел, разделенных пробелами (по одному пробелу между числами). Какие цифры присутствуют ровно в одном числе дважды?

22. С клавиатуры вводится несколько строк, последняя строка – пустая (пустая строка – признак окончания ввода и дальше игнорируется). Выведите символы, что присутствуют в каждой строке. Предполагается, что коды всех символов в строке не превышают 127.

23. С клавиатуры вводится несколько строк, последняя строка – пустая (пустая строка – признак окончания ввода и дальше игнорируется). Выведите символы, что присутствуют ровно в одной строке. Предполагается, что коды всех символов в строке не превышают 127.

24. С клавиатуры вводится несколько строк, последняя строка – пустая (пустая строка – признак окончания ввода и дальше игнорируется). Выведите символы, что присутствуют ровно в двух строках. Предполагается, что коды всех символов в строке не превышают 127.

25. С клавиатуры вводится несколько строк, последняя строка – пустая (пустая строка – признак окончания ввода и дальше игнорируется). Выведите символы, что отсутствуют ровно в двух строках. Предполагается, что коды всех символов в строке не превышают 127.

Задание №3

Обратите внимание, что в этом задании требуется оптимальное решение (продвинутые функции дадут менее оптимальные решения из-за копирования). Можно предполагать, что все символам соответствует одно значение типа char (однако, если вы сделаете корректное решение (разумеется, вне пары), то это будет плюсом).

1. Найдите первый символ в первом максимально длинном слове с нечетным числом символов в строке (в строке указываются только слова, разделенные одним или несколькими пробелами).

2. Найдите последний символ в первом максимально длинном слове с нечетным числом символов в строке (в строке указываются только слова, разделенные одним или несколькими пробелами).

3. Найдите первый символ в последнем максимально длинном слове с нечетным числом символов в строке (в строке указываются только слова, разделенные одним или несколькими пробелами).

4. Найдите последний символ в последнем максимально длинном слове с нечетным числом символов в строке (в строке указываются только слова, разделенные одним или несколькими пробелами).

5. Найдите первый символ в первом самом коротком слове в строке с нечетным числом символов (в строке указываются только слова, разделенные одним или несколькими пробелами).

6. Найдите последний символ в первом самом коротком слове в строке с нечетным числом символов (в строке указываются только слова, разделенные одним или несколькими пробелами).

7. Найдите первый символ в последнем самом коротком слове в строке с нечетным числом символов (в строке указываются только слова, разделенные одним или несколькими пробелами).

8. Найдите последний символ в последнем самом коротком слове в строке с нечетным числом символов (в строке указываются только слова, разделенные одним или несколькими пробелами).

9. Найдите первый символ в первом максимально длинном слове с четным числом символов в строке (в строке указываются только слова, разделенные одним или несколькими пробелами).

10. Найдите последний символ в первом максимально длинном слове с четным числом символов в строке (в строке указываются только слова, разделенные одним или несколькими пробелами).

11. Найдите первый символ в последнем максимально длинном слове с четным числом символов в строке (в строке указываются только слова, разделенные одним или несколькими пробелами).

12. Найдите последний символ в последнем максимально длинном слове с четным числом симвлов в строке (в строке указываются только слова, разделенные одним или несколькими пробелами).

13. Найдите первый символ в первом самом коротком слове в строке с четным числом символов (в строке указываются только слова, разделенные одним или несколькими пробелами).

14. Найдите последний символ в первом самом коротком слове в строке с четным числом символов (в строке указываются только слова, разделенные одним или несколькими пробелами).

15. Найдите первый символ в последнем самом коротком слове в строке с четным числом символов (в строке указываются только слова, разделенные одним или несколькими пробелами).

16. Найдите последний символ в последнем самом коротком слове в строке с четным числом символов (в строке указываются только слова, разделенные одним или несколькими пробелами).

17. Найдите второй символ в первом максимально длинном слове с четным числом символов в строке (в строке указываются только слова, разделенные одним или несколькими пробелами).

18. Найдите предпоследний символ в первом максимально длинном слове с четным числом символов в строке (в строке указываются только слова, разделенные одним или несколькими пробелами).

19. Найдите второй символ в последнем максимально длинном слове с четным числом символов в строке (в строке указываются только слова, разделенные одним или несколькими пробелами).

20. Найдите предпоследний символ в последнем максимально длинном слове с четным числом символов в строке (в строке указываются только слова, разделенные одним или несколькими пробелами).

21. Найдите второй символ в первом самом коротком слове в строке с четным числом символов (в строке указываются только слова, разделенные одним или несколькими пробелами).

22. Найдите предпоследний символ в первом самом коротком слове в строке с четным числом символов (в строке указываются только слова, разделенные одним или несколькими пробелами).

23. Найдите второй символ в последнем самом коротком слове в строке с четным числом символов (в строке указываются только слова, разделенные одним или несколькими пробелами).

24. Найдите предпоследний символ в последнем самом коротком слове в строке с четным числом символов (в строке указываются только слова, разделенные одним или несколькими пробелами).

25. Найдите первый символ в первом максимально длинном слове с числом символов,кратным трем, в строке (в строке указываются только слова, разделенные одним или несколькими пробелами).

2.2 Практическая работа «функции Kotlin» (2 часа)

Задание №1

Измените программу, сделанную в задании №1 практической работы №1: основной алгоритм вынесите в функцию. Опишите функцию несколькими способами:

1. как обычную функцию;
2. как tailrec-функцию.

Обратите внимание на то, что основной алгоритм не должен содержать ввод-вывод и работу с любыми глобальными самописными объектами.

Задание №2

В программах, сделанных в задании №1, вынесите проверяемое условие (то есть то условие, что сформулировано в задании) в отдельную single-expression функцию.

Задание №3

Функции, созданные в задании №1, модифицируйте таким образом, чтобы условие, по которому происходит отбор, можно было передавать как аргумент (один из аргументов функции должен быть функционального типа со значением по умолчанию – условием, что указано было в вашем варианте).

Задание №4

В задании №3 первой практической работы реализуйте следующее: выделите основной алгоритм в отдельную функцию (без ввода и вывода, использования глобальных переменных), куда в качестве аргумента передавайте функцию, имеющую смысл – способ сравнения двух чисел; изменяя данную функцию, пользователь вашей функции (то есть другой программист) должен иметь возможность получить информацию либо о самых длинных, либо о самых коротких словах. Аналогичным образом передавайте в вашу функцию функцию, которая будет определять условие отбора слов.

Задание №5

Создайте функцию, которая реализует алгоритм второго задания первой практической работы, в которую все числа, слова или пары (в зависимости от варианта) передаются в аргументах функции. Например: f(123,25,222); f("dfd "dd "ddd"); d (Pair(2,3),Pair(3,5),Pair(4,1)). В этом задании вывод может осуществлять во вновь создаваемой функции.

Задание №6

1. Создайте функцию, которая по данным функциям с параметром типа Int и результатами типа Int возвращает новую функцию – сумму данных (количество исходных функций – любое).
2. Создайте функцию, которая по данным функциям с параметром типа Int и результатами типа Int возвращает новую функцию – произведение данных (количество исходных функций – любое).
3. Создайте функцию, которая по данным функциям с параметром типа Int и результатами типа Int возвращает новую функцию – максимум данных (количество исходных функций – любое).
4. Создайте функцию, которая по данным функциям с параметром типа Int и результатами типа Int возвращает новую функцию – минимум данных (количество исходных функций – любое).
5. Создайте функцию, которая по данной функции $f : Int -> Int$ и числу n возвращает функцию $f(f(f(...f(x)...)$, где f вызывается n раз.

6. Создайте функцию, которая по данным функциям без параметров и результатам типа String возвращает новую функцию без параметров, что возвращает конкатенацию данных (количество исходных функций – любое).

7. Создайте функцию, которая по данным функциям с параметром типа Int и результатами типа Int возвращает новую функцию с аргументом x типа Int, которая возвращает номер первой функции, имеющей максимальное значение, при подстановке в качестве аргумента x. (количество исходных функций – любое).

8. Создайте функцию, которая по данным функциям с параметром типа Int и результатами типа Int возвращает новую функцию с аргументом x типа Int, которая возвращает номер первой функции, имеющей минимальное значение, при подстановке в качестве аргумента x. (количество исходных функций – любое).

9. Создайте функцию, которая по данным функциям с параметром типа Int и результатами типа Int возвращает новую функцию с аргументом x типа Int, которая возвращает номер последней функции, имеющей максимальное значение, при подстановке в качестве аргумента x. (количество исходных функций – любое).

10. Создайте функцию, которая по данным функциям с параметром типа Int и результатами типа Int возвращает новую функцию с аргументом x типа Int, которая возвращает номер последней функции, имеющей минимальное значение, при подстановке в качестве аргумента x. (количество исходных функций – любое).

11. Создайте функцию, которая по данным двум функциям с параметром типа Int и результатами типа Int? возвращает новую функцию – сумму данных. Если результат хотя бы одной из суммируемых функций – null, то и результат возвращаемой функции – null.

12. Создайте функцию, которая по данным двум функциям с параметром типа Int и результатами типа Int? возвращает новую функцию – произведение данных. Если результат хотя бы одной из умножаемых функций – null, то и результат возвращаемой функции – null.

13. Создайте функцию, которая по данным двум функциям с параметром типа Int и результатами типа Int? возвращает новую функцию – максимум данных. Если результат хотя бы одной из исходных функций – null, то и результат возвращаемой функции – null.

14. Создайте функцию, которая по данным двум функциям с параметром типа Int и результатами типа Int? возвращает новую функцию – минимум данных. Если результат хотя бы одной из исходных функций – null, то и результат возвращаемой функции – null.

15. Создайте функцию, которая по двум данным функциям f(x) и g(x) возвращает функцию f(g(x)), параметры всех упомянутых функций имеют тип Int, результат – Int?. Если функция g для данного x дает результат null, то результирующая функция так же равна null.

16. Создайте функцию, которая по данной функции с параметром типа Int и результатом типа Int, а также целому числу n возвращает новую функцию, которая по массиву из n элементов типа Int возвращает массив результатов применения функции f к каждому элементу данного массива.

17. Создайте функцию, которая по данному массиву целых чисел возвращает функцию, которая при каждом вызове последовательно возвращает элементы массива, а когда элементы кончатся – null.

18. Создайте функцию, которая по данной функции, имеющей аргумент типа Int и результат типа Int, возвращает функцию, которая при каждом вызове последовательно возвращает результаты применения функции-аргумента к числам 1, 2, 3, ...

19. Создайте функцию, которая по данному массиву целых чисел возвращает функцию, которая при каждом вызове последовательно возвращает элементы массива в обратном порядке, а когда элементы кончатся – null.

20. Создайте функцию, которая по данной строке возвращает функцию, которая при каждом вызове последовательно возвращает символы строки, а когда символы кончатся – null.

21. Создайте функцию, которая по данной строке возвращает функцию, которая при каждом вызове последовательно возвращает символы строки в обратном порядке, а когда символы кончатся – null.

22. Создайте функцию, которая по данным функциям с параметром типа Float и результатами типа Float возвращает новую функцию – среднее арифметическое данных (количество исходных функций – любое).

23. Создайте функцию, которая по данным функциям с параметром типа Float и результатами типа Float возвращает новую функцию – среднее квадратическое данных (количество исходных функций – любое).

24. Создайте функцию, которая по данным функциям с параметром типа Float и результатами типа Float возвращает новую функцию – среднее геометрическое данных (количество исходных функций – любое).

25. Создайте функцию, которая по данной функции $f : Float- > Float$ и числу x возвращает функцию, которая при каждом вызове последовательно возвращает $f(x)$, $f(f(x))$, $f(f(f(x)))$, ...

2.3 Практическая работа «Функции высшего порядка» (4 часа)

В данной работе требуется написать не самую оптимальную реализацию, а реализацию, которая наиболее полноценно использует функции над коллекциями, использующие функциональный подход, и строковые функции. В работе запрещено использовать mutable коллекции и var переменные.

Реализация должна состоять из одной строки с точечными вызовами, включая ввод и вывод, использовать рекурсию запрещено.

Примечание: данный способ реализации программы нужен исключительно в учебных целях, в дальнейшем разбивайте подобные решения на небольшие функции, которые удобно повторно использовать.

Задания №№1-3 Реализуйте задания №№1-3 первой практической работы.

Задание №4 С клавиатуры вводится несколько целых значений через пробел. Найдите (без учета тех чисел, где соответствующей цифры нет):

1. Побитовое И предпоследней цифры всех чисел
2. Побитовое ИЛИ предпоследней цифры всех чисел
3. Побитовое исключающее ИЛИ предпоследней цифры всех чисел
4. Побитовый штрих Шеффера последней цифры всех чисел (операции выполняются слева направо)
5. Побитовый штрих Шеффера предпоследней цифры всех чисел (операции выполняются слева направо)
6. Побитовую стрелку Пирса последней цифры всех чисел (операции выполняются слева направо)
7. Побитовую стрелку Пирса предпоследней цифры всех чисел (операции выполняются слева направо)
8. Побитовый штрих Шеффера последней цифры всех чисел (операции выполняются справа налево)
9. Побитовый штрих Шеффера предпоследней цифры всех чисел (операции выполняются справа налево)
10. Побитовую стрелку Пирса последней цифры всех чисел (операции выполняются справа налево)
11. Побитовую стрелку Пирса предпоследней цифры всех чисел (операции выполняются справа налево)
12. Побитовое И первой цифры всех чисел
13. Побитовое ИЛИ первой цифры всех чисел
14. Побитовое исключающее ИЛИ первой цифры всех чисел
15. Побитовое И второй цифры всех чисел
16. Побитовое ИЛИ второй цифры всех чисел
17. Побитовое исключающее ИЛИ второй цифры всех чисел

18. Побитовый штрих Шеффера первой цифры всех чисел (операции выполняются слева направо)
19. Побитовый штрих Шеффера второй цифры всех чисел (операции выполняются слева направо)
20. Побитовую стрелку Пирса первой цифры всех чисел (операции выполняются слева направо)
21. Побитовую стрелку Пирса второй цифры всех чисел (операции выполняются слева направо)
22. Побитовый штрих Шеффера первой цифры всех чисел (операции выполняются справа налево)
23. Побитовый штрих Шеффера второй цифры всех чисел (операции выполняются справа налево)
24. Побитовую стрелку Пирса первой цифры всех чисел (операции выполняются справа налево)
25. Побитовую стрелку Пирса второй цифры всех чисел (операции выполняются справа налево)

Задание №5

1. С клавиатуры вводится информация о студентах: фамилия, имя, оценки. Выведите на экран информацию о трех лучших студентах по среднему баллу. В случае, если у нескольких студентов средний балл совпадает, то выведите большее число студентов (пока не будут выведены все студенты или не будут полностью исчерпаны студенты с тремя лучшими баллами). Вывод надо осуществлять в порядке убывания среднего балла, а для одинаковых средних баллов – в алфавитном порядке по фамилии и имени.
2. С клавиатуры вводится информация о студентах: фамилия, имя, оценки. Выведите на экран информацию о трех лучших студентах по максимальному баллу. В случае, если у нескольких студентов средний балл совпадает, то выведите большее число студентов (пока не будут выведены все студенты или не будут полностью исчерпаны студенты с тремя лучшими баллами). Вывод надо осуществлять в порядке убывания максимального балла, а для одинаковых максимальных баллов – в алфавитном порядке по фамилии и имени.
3. С клавиатуры вводится информация о студентах: фамилия, имя, оценки. Выведите на экран информацию о трех лучших студентах по минимальному баллу. В случае, если у нескольких студентов средний балл совпадает, то выведите большее число студентов (пока не будут выведены все студенты или не будут полностью исчерпаны студенты с тремя лучшими баллами). Вывод надо осуществлять в порядке убывания минимального балла, а для одинаковых минимальных баллов – в алфавитном порядке по фамилии и имени.
4. С клавиатуры вводится информация о студентах: фамилия, имя, оценки. Выведите на экран информацию о трех худших студентах по среднему баллу. В случае, если у нескольких студентов средний балл совпадает, то выведите большее число студентов (пока не будут выведены все студенты или не будут полностью исчерпаны студенты с тремя худшими баллами). Вывод надо осуществлять в порядке возрастания среднего балла, а для одниковых средних баллов – в алфавитном порядке по фамилии и имени.

5. С клавиатуры вводится информация о студентах: фамилия, имя, оценки. Выведите на экран информацию о трех худших студентах по максимальному баллу. В случае, если у нескольких студентов максимальный балл совпадает, то выведите большее число студентов (пока не будут выведены все студенты или не будут полностью исчерпаны студенты с тремя худшими баллами). Вывод надо осуществлять в порядке возрастания максимального балла, а для одинаковых максимальных баллов – в алфавитном порядке по фамилии и имени.

6. С клавиатуры вводится информация о студентах: фамилия, имя, оценки. Выведите на экран информацию о трех худших студентах по минимальному баллу. В случае, если у нескольких студентов минимальный балл совпадает, то выведите большее число студентов (пока не будут выведены все студенты или не будут полностью исчерпаны студенты с тремя минимальными баллами). Вывод надо осуществлять в порядке возрастания минимального балла, а для одинаковых минимальных баллов – в алфавитном порядке по фамилии и имени.

7. С клавиатуры вводится информация об абитуриентах: фамилия, имя, а далее названия предметов и оценки ЕГЭ по ним. Выведите на экран информацию о трех лучших абитуриентах по максимальному баллу за сумму трех ЕГЭ. В случае, если у нескольких абитуриентов средний балл совпадает, то выведите большее число абитуриентов (пока не будут выведены все абитуриенты или не будут полностью исчерпаны абитуриентами с тремя лучшими баллами). Вывод надо осуществлять в порядке убывания максимальной суммы баллов за три ЕГЭ, а для одинаковых сумм баллов – в алфавитном порядке по фамилии и имени.

8. С клавиатуры вводится информация об абитуриентах: фамилия, имя, а далее названия предметов и оценки ЕГЭ по ним. Выведите на экран информацию о трех худших абитуриентах по максимальному баллу за сумму трех ЕГЭ. В случае, если у нескольких абитуриентов средний балл совпадает, то выведите большее число абитуриентов (пока не будут выведены все абитуриенты или не будут полностью исчерпаны абитуриентами с тремя худшими баллами). Вывод надо осуществлять в порядке возрастания максимальной суммы баллов за три ЕГЭ, а для одинаковых сумм баллов – в алфавитном порядке по фамилии и имени.

Задание №6

1. По номеру числа Фибоначчи найдите число Фибоначчи (не используйте факты, которые вы не можете доказать самостоятельно)

2. По числу Фибоначчи найдите его номер (не используйте факты, которые вы не можете доказать самостоятельно)

3. По натуральному числу найдите его факториал

4. По факториалу найдите исходное число

5. По данному числу найдите простое число с таким номером (если простые числа нумеровать в порядке возрастания)

6. По простому числу определите его номер в последовательности всех простых чисел, расположенных по возрастанию

7. По данному числу найдите все его простые делители

8. По числу n найдите n-ое совершенное число (не используйте факты, которые вы не можете доказать самостоятельно)

9. По совершенному числу найдите его номер в последовательности всех совершенных чисел, расположенных в порядке возрастания (не используйте факты, которые вы не можете доказать самостоятельно)

10. По натуральному числу найдите его двойной факториал

11. По двойному факториалу найдите исходное число

12. Рассмотрим все тройки натуральных чисел, удовлетворяющих уравнению $a^2+b^2=c^2$. Для данного n найдите такую тройку чисел a, b, c, что $a^2+b^2=c^2$, чтобы $a+b+c$ было меньше n и наиболее близко к n.

13. Рассмотрим все тройки натуральных чисел, удовлетворяющих уравнению $a^2+b^2=c^2$. Для данного n найдите такую тройку чисел a, b, c, что $a^2+b^2=c^2$, чтобы $a+b+c$ было больше n и наиболее близко к n.

14. По данному натуральному числу n найдите наименьшее простое число, большее n

15. По данному натуральному числу n найдите наибольшее простое число, меньшее n

16. По данному натуральному числу n найдите наименьший факториал, больший n

17. По данному натуральному числу n найдите наибольший факториал, меньший n

18. По данному натуральному числу n найдите наименьший двойной факториал, больший n

19. По данному натуральному числу n найдите наибольший двойной факториал, меньший n

20. По данному натуральному числу n найдите наименьшее число Фибоначчи, большее n (не используйте факты, которые вы не можете доказать самостоятельно)

21. По данному натуральному числу n найдите наибольшее число Фибоначчи, меньшее n (не используйте факты, которые вы не можете доказать самостоятельно)

22. По данному натуральному числу n найдите наименьшее совершенное число, большее n (не используйте факты, которые вы не можете доказать самостоятельно)

23. По данному натуральному числу n найдите наибольшее совершенное число, меньшее n (не используйте факты, которые вы не можете доказать самостоятельно)

24. Для данного натурального числа n найдите такое простое число p, что входит в разложение на простые множители числа n наибольшее число раз.

25. Для данного натурального числа n найдите такое простое число p, что входит в разложение на простые множители числа n наименьшее число раз.

2.4 Практическая работа «Регулярные выражения» (2 часа)

Используя функции, реализующие работу с регулярными выражениями, разработайте программу проверки синтаксической корректности фрагмента программы на языке Паскаль (классический диалект) в соответствии с вашим вариантом (номер варианта в данной работе отличается от предыдущих).

Указание: проверьте соответствие вашего представления об языке стандарту языка.

В заданиях нигде не предполагается, что можно использовать необязательные круглые скобки и составные операторы без явного указания на это.

1 Программа проверки правильности оператора case с переменной-селектором типа char (без else), в который вложены операторы присваивания переменной строкового литерала.

2 Программа проверки правильности оператора присваивания, в правой части которого допустимы операции сложения, вычитания, умножения, деления, переменные, целые и вещественные числа (включая показательную форму).

3 Программа проверки правильности описания массива, у которого индексы могут иметь тип-название (например, boolean), ограниченный тип (для integer), а тип-элемента – название типа (идентификатор).

4 Программа проверки правильности описания массива, у которого индексы могут иметь тип-название (например, boolean), перечислимый тип, а тип-элемента – ограниченный тип для integer.

5 Программа проверки правильности описания массива, у которого индексы могут иметь тип-название (например, boolean), ограниченный тип (для char), а тип-элемента – перечислимый тип.

6 Программа проверки описания перечислимого и ограниченного типа (для integer, char). Строка должна начинаться со слова type.

7 Программа проверки правильности оператора write(ln), у которого в качестве аргументов могут участвовать строковый литералы, целые и вещественные числа, выражения над числами, соединенные операциями сложения, вычитания, умножения и деления. Может также быть указан формат вывода.

8 Проверка правильности последовательности операторов read(ln), присваивания и write(ln). У writeln возможные аргументы – строковые литералы; у read(ln) – названия переменных, элементы массивов. У оператора присваивания слева название переменной или элемент массива, справа число (целое или вещественное).

9 Проверка правильности оператора for, у которого начальным и конечным значениями могут быть как целые числа, так и символы, а тело цикла – оператор write(ln), у которого аргументы – целые и вещественные числа и переменные.

10 Проверка правильности оператора if, у которого условие имеет вид: `<переменная><знак><число>` , при этом знак – это знак больше, меньше, больше или равно, меньше или равно, равно, не равно; число – целое или вещественное. Как в части then, так и в else (может быть опущен) указывается оператор write(ln) с аргументами – переменными и строками.

11 Проверка правильности последовательности операторов присваивания, правая часть которых – выражения, в которых используются литералы типа «множество» со значениями типа integer, переменные и операции +,-,*.

12 Проверка правильности последовательности операторов присваивания, правая часть которых – выражения, в которых используются литералы типа «множество» со значениями типа char, переменные и операции +,-,*.

13 Проверка правильности оператора while, у которого условие имеет вид: `<переменная><знак><число>` , при этом знак – это знак больше, меньше, больше или равно, меньше или равно, равно, не равно; число – целое или вещественное. Телом цикла является либо снова оператор while, либо оператор присваивания, в правой части которого могут быть представлены целые числа, либо переменные.

14 Проверка правильности оператора if, у которого условие имеет вид: `<переменная><знак><строка>` , при этом знак – это знак больше, меньше, больше или равно, меньше или равно, равно, не равно. Как в части then, так и в else (может быть опущен) указывается оператор присваивания, правая часть которого – выражение, содержащее переменные, целые и вещественные числа и знаки операций +,-,*,/.

15 Проверка правильности оператора repeat..until, у которого условие имеет вид: `<переменная><знак><числ` при этом знак – это знак больше, меньше, больше или равно, меньше или равно, равно, не равно; число – целое или вещественное. Телом цикла является либо снова оператор repeat..until, либо оператор присваивания, в правой части которого могут быть представлены целые числа, либо переменные.

16 Программа проверки правильности оператора case с переменной-селектором перечислимого типа (без else), в который вложены операторы присваивания переменной константы типа integer.

17 Проверка правильности последовательности операторов присваивания, правая часть которых – выражения, в которых используются литералы типа «множество» со значениями перечислимого типа, переменные и операции +, -, *.

18 Проверка правильности оператора for, у которого начальным и конечным значениями могут быть значения перечислимого типа, а тело цикла – оператор присваивания, в правой части которого записана числовая константа.

19 Проверка правильности оператора if, у которого условие – это логическое выражение над переменными логического типа. Как в части then, так и в else (может быть опущен) указывается оператор write(ln) с аргументами – переменными и числами.

20 Проверка правильности оператора while, у которого условие – это логическое выражение над переменными логического типа. Телом цикла является оператор присваивания, правая часть которого – выражение, содержащее целые константы и операции сложения, вычитания, умножения и деления.

21 Проверка правильности оператора repeat..until, у которого условие – это логическое выражение над переменными логического типа. Телом цикла является оператор присваивания, правая часть которого – выражение, содержащее целые константы и операции сложения, вычитания, mod и div.

22 Проверка правильности оператора repeat..until, у которого условие имеет вид: `<переменная><знак><строк` при этом знак – это знак больше, меньше, больше или равно, меньше или равно, равно, не равно; строка – литерал типа string. Телом цикла является оператор writeln, аргументами которого являются целые и вещественные числа, возможно, с указанием формата.

23 Проверка правильности оператора while, у которого условие имеет вид: `<переменная><знак><строка>` , при этом знак – это знак больше, меньше, больше или равно, меньше или равно, равно, не равно; строка – литерал типа string. Телом цикла является либо снова оператор while, либо оператор writeln, аргументами которого являются выражения, содержащие целые числа, имена переменных и операции +, -, *. /.

24 Программа проверки правильности оператора write(ln), у которого в качестве аргументов могут участвовать строковый литералы и переменные, соединенные операцией сложения; целые числа и переменные, соединенные операциями сложения, вычитания, mod и div. Может также быть указан формат вывода.

25 Проверка правильности последовательности операторов присваивания, правая часть которых – выражения, в которых используются строковые литералы и переменные, соединенные знаком `+`; а также выражения, в которых используются литералы типа integer, переменные и операции сложения, вычитания, `div` и `mod`.

2.5 Практическая работа «Особенности ООП в Kotlin» (12 часов)

Реализуйте с использованием ООП простейшую консольную базу данных (без красивого интерфейса) в соответствии со своим вариантом. Функции: добавление, изменение, удаление, сортировка, поиск, вывод на экран содержимого. База данных не предполагает сохранении информации между сеансами работы, но подразумевает, что программа не «падает» при (почти) любых действиях пользователя. Исключение: переполнение памяти.

Осуществите распределение ваших классов, объектов и интерфейсов по различным файлам, а файлы сгруппируйте по папкам по какому-либо принципу.

Обратите внимание на то, что вам следует разделить ваш проект на Model и View, где Model будет без изменений перенесена в проект на Android, а View содержит логику консольного взаимодействия.

В ходе реализации строго используйте следующие принципы:

- DRY
- SOLID

1. База данных студентов группы. Поля: фамилия, имя, отчество, пол, возраст.
2. База данных расходов семьи. Поля: товар, стоимость, количество, дата.
3. База данных загрузки аудиторий. Поля: дата и время, начала, дата и время конца, аудитория, преподаватель.
4. База данных учета доходов и расходов предпринимателя. Поля: дата, тип операции (доход/расход), объем операции, описание, корреспондент.
5. База данных велоклуба. Поля: ФИО, тип велосипеда (MTB и др.), стаж участия в велоклубе.
6. База данных рейсов авиакомпании. Поля: дата и время вылета, аэропорт вылета, аэропорт прилета, дата и время прилета, марка самолета.
7. База данных автобусных маршрутов. Поля: номер маршрута, номер парка, времена начала и окончания движения, длина маршрута в км.
8. База данных электричек. Поля: вокзал, номер поезда, количество вагонов, тип (экспресс/обычный/спутник).
9. База данных товаров Интернет-магазина. Поля: название товара, категория, цена товара, описание товара.
10. База заказов Интернет-магазина. Поля: ФИО заказчика, стоимость заказа, скидка (в процентах), адрес доставки.
11. База данных выборов. Поля: участок, кандидат, количество голосов.
12. База данных практических работ. Поля: практическая работа, студент, номер варианта, номер уровня, дата сдачи, оценка.
13. База данных операторов и телеканалов. Поля: Название, тип (спутник, кабель, Интернет), охват (кол-во миллионов домохозяйств), минимальная стоимость подписки.
14. База данных тарифных планов оператора. Поля: название, тип вещания (обычный/HD), флаг общедоступности.

15. База данных незаконно огороженных берегов. Поля: водный объект, регион, GPS-координаты, длина недоступного участка берега, дата фиксации нарушения.

16. База данных временного прекращения движения в метро. Поля: дата и время начала прекращения движения, дата и время окончания прекращения движения, станция, станция (от какой до какой станции прекращено движение).

17. База данных проката фильмов. Поля: дата, время, кинотеатр, фильм, номер зала, тип сеанса (3D/Imax/обычный).

18. База данных эвакуированных автомобилей. Поля: улица, автостоянка, GPS-координаты, тип нарушения (стоянка на проезжей части в месте запрета, стояна на тротуаре, стоянка на газоне), номер автомобиля, тип автомобиля (легковой/грузовой малой тонажности/грузовой большой тонажности).

19. База данных средних специальных учебных учреждений. Поля: название, адрес, тип подчинения (федеральный/региональный), год основания, номер лицензии, номер аккредитации, дата окончания действия аккредитации.

20. База данных поселков. Поля: название, девелопер, площадь, количество жителей.

21. База данных сухопутной военной техники. Поля: название, модель, разработчик, предприятие, стоимость, тип.

22. База данных деревьев в городе. Поля: GPS-координаты, вид дерева, округ, год посадки.

23. База данных футбольных матчей. Поля: дата, команда, команда, счет, место проведения.

24. База данных обращений жителей. Поля: дата, время, объект, заявитель, содержание обращения (до 255 символов), дата ответа, ответ на обращение (до 255 символов).

25. База данных студентов колледжа. Поля: ФИО, группа, признак бюджетности, стипендия (нет/обычная/повышенная), флаг наличия социальной стипендии, дата рождения.

2.6 Практическая работа «Dagger» (2 часа)

Реализуйте DI в проекте, реализованном в последней практической работе, с использованием Dagger.

2.7 Практическая работа «Абстрактное программирование в Kotlin» (2 часа)

Задание №1

1. Создайте функцию, которая по данным функциям с параметром любого типа и результатами типа Int возвращает новую функцию – сумму данных (количество исходных функций – любое).

2. Создайте функцию, которая по данным функциям с параметром любого типа и результатами типа Int возвращает новую функцию – произведение данных (количество исходных функций – любое).

3. Создайте функцию, которая по данным функциям с параметром любого типа и результатами типа Int возвращает новую функцию – максимум данных (количество исходных функций – любое).

4. Создайте функцию, которая по данным функциям с параметром любого типа и результатами типа Int возвращает новую функцию – минимум данных (количество исходных функций – любое).

5. Создайте функцию, которая по данной функции $f : T- > T$ и числу n возвращает функцию $f(f(f(...f(x)...)$, где f вызывается n раз. Здесь T – любой тип.

6. Создайте функцию, которая по данным функциям с единственным параметром типа T и результатами типа String возвращает новую функцию с параметром типа T, что возвращает конкатенацию данных (количество исходных функций – любое).

7. Создайте функцию, которая по данным функциям с параметром типа T и результатами типа Int возвращает новую функцию с аргументом x типа T, которая возвращает номер первой функции, имеющей максимальное значение, при подстановке в качестве аргумента x. (количество исходных функций – любое). Здесь T – любой тип.

8. Создайте функцию, которая по данным функциям с параметром типа T и результатами типа Int возвращает новую функцию с аргументом x типа T, которая возвращает номер первой функции, имеющей минимальное значение, при подстановке в качестве аргумента x. (количество исходных функций – любое). Здесь T – любой тип.

9. Создайте функцию, которая по данным функциям с параметром типа T и результатами типа Int возвращает новую функцию с аргументом x типа T, которая возвращает номер последней функции, имеющей максимальное значение, при подстановке в качестве аргумента x. (количество исходных функций – любое). Здесь T – любой тип.

10. Создайте функцию, которая по данным функциям с параметром типа T и результатами типа Int возвращает новую функцию с аргументом x типа T, которая возвращает номер последней функции, имеющей минимальное значение, при подстановке в качестве аргумента x. (количество исходных функций – любое). Здесь T – любой тип.

11. Создайте функцию, которая по данным двум функциям с параметром типа T и результатами типа Int? возвращает новую функцию – сумму данных. Если результат хотя бы одной из суммируемых функций – null, то и результат возвращаемой функции – null. Здесь T – любой тип.

12. Создайте функцию, которая по данным двум функциям с параметром типа T и результатами типа Int? возвращает новую функцию – произведение данных. Если результат хотя бы одной из умножаемых функций – null, то и результат возвращаемой функции – null. Здесь T – любой тип.

13. Создайте функцию, которая по данным двум функциям с параметром типа T и результатами типа Int? возвращает новую функцию – максимум данных. Если результат хотя бы одной из исходных функций – null, то и результат возвращаемой функции – null. Здесь T – любой тип.

14. Создайте функцию, которая по данным двум функциям с параметром типа T и результатами типа Int? возвращает новую функцию – минимум данных. Если результат хотя бы одной из исходных функций – null, то и результат возвращаемой функции – null. Здесь T – любой тип.

15. Создайте функцию, которая по двум данным функциям f(x) и g(x) возвращает функцию f(g(x)), параметры всех упомянутых функций имеют тип T, результат – T?. Если функция g для данного x дает результат null, то результирующая функция так же равна null. Здесь T – любой тип.

16. Создайте функцию, которая по данной функции с параметром типа T и результатом типа Int, а также целому числу n возвращает новую функцию, которая по массиву из n элементов типа T возвращает массив результатов применения функции f к каждому элементу данного массива. Здесь T – любой тип.

17. Создайте функцию, которая по данному массиву значений типа T возвращает функцию, которая при каждом вызове последовательно возвращает элементы массива, а когда элементы кончатся – null. Здесь T – любой тип.

18. Создайте функцию, которая по данной функции, имеющей аргумент типа Int и результат произвольного типа, возвращает функцию, которая при каждом вызове последовательно возвращает результаты применения функции-аргумента к числам 1, 2, 3, ...

19. Создайте функцию, которая по данному массиву значений произвольного типа возвращает функцию, которая при каждом вызове последовательно возвращает элементы массива в обратном порядке, а когда элементы кончатся – null.

Задание №2
Обозначим n – номер варианта.
Функция добавления элемента в список выбирается учащимся исходя из значения `n % 4+1`

1. `fun push (el: T): Bool` вставляет элемент в начало списка;

2. `fun add (el: T): Bool` вставляет элемент в конец списка;

3. `fun insert (el: T,n: Int): Bool` вставляет элемент на позицию n (нумерация идет с единицы) с начала списка;

4. `fun insert (els:Array<T>,count: Int,n: Int): Bool` вставляет count элементов массива els начиная с позиции n

Процедура удаления элемента из списка выбирается учащимся исходя из значения `n/4%4+1`:

1. `fun delete(): Bool` удаляет элемент из начала списка
2. `fun delete(): Bool` удаляет элемент с конца списка
3. `fun delete (n: Int): Bool` удаляет элемент с позиции n
4. `fun delete (count: Int,n: Int): Bool` удаляет count элементов начиная с позиции n

Процедура печати элементов списка выбирается учащимся исходя из значения `n mod 5+1`:

1. `fun print():Unit` печатает все элементы
2. `fun print():Unit` печатает первый элемент списка
3. `fun print():Unit` печатает последний элемент списка
4. `fun print(n: Int): Unit` печатает элемент списка, имеющий позицию n
5. `fun print(count: Int,n: Int): Unit` печатает count элементов списка, начиная с позиции n

Кроме того должна быть реализована функция eraseAll, которая очищает весь список. Список должен быть реализован в виде generic-класса.

2.8 Практическая работа «Простейшие программы для Android» (2 часа)

Разработайте программу, работающую под управлением Android с использованием Views. Проверьте, что программа корректно работает с различными размерами экрана, а также при повороте экрана.

1. Программа решения квадратного уравнения
2. Программа решения неравенства вида $ax + b > 0$
3. Программа решения неравенства вида $ax + b < 0$
4. Программа решения неравенства вида $ax + b \geqslant 0$
5. Программа решения неравенства вида $ax + b \leqslant 0$
6. Программа поиска дня недели по числу и месяцу в текущем году
7. Программа перевода числа из 10-ой в 16-ую, 8-ую и 2-ую систем.
8. Программа поиска времени, когда окончится интервал. Дано: часы и минуты начала интервала и количество минут, сколько он идет. Результат: часы и минуты окончания интервала.
9. Программа поиска обратной матрицы для матрицы 3×3.
10. Программа поиска длины интервала. Дано: часы и минуты начала интервала и часы и минуты конца интервала. Результат: количество минут в интервале.
11. Программа умножения и деления двух комплексных чисел.
12. Программа нахождения площади треугольника по координатам вершин.
13. Программа нахождения углов треугольника по координатам вершин (проще всего это сделать по теореме косинусов).
14. Программа перевода числа из 16-ой, 8-ой и 2-ой системы в 10-ую систему счисления.
15. Программа нахождения количества денег на вкладе после окончания его срока по начальному взносу, проценту и срока в годах.
16. Программа нахождения степени комплексного числа. Исходные данные: действительная, мнимая часть числа и степень. Результат: действительная и мнимая часть результата.
17. Программа умножения и деления чисел, представленных в виде обыкновенных дробей (состоящих из целой части, числителя и знаменателя). Не забудьте выполнить сокращение дроби и приведение ее к правильному виду.
18. Программа сложения и вычитания чисел, представленных в виде обыкновенных дробей (состоящих из целой части, числителя и знаменателя). Не забудьте выполнить сокращение дроби и приведение ее к правильному виду.
19. Программа определения по дате (число и месяц) знака зодиака.

20. Программа определения по обыкновенной дроби (числителю и знаменателю) периода десятичной дроби.

21. Программа перевода комплесного числа из обычной формы в тригонометрическую и наоборот.

22. Программа-игра Баше. При реализации этого задания не требуется ничего рисовать, вся информация вводится и выводится в виде чисел в обычные элементы управления.

23. Программа разложения числа на простые множители.

24. Программа нахождения наибольшего общего делителя и наименьшего общего кратного двух натуральных чисел.

25. Программа-тест по предмету «Разработка мобильных приложений». Создайте программу-тест из 10 вопросов с выбором вариантов ответов и показом результатов прохождения теста.

2.9 Практическая работа «Jetpack Compose» (2 часа)

Перепишите программу, реализованную в прошлой практической работе с использованием Jetpack Compose

2.10 Практическая работа «Автоматизация тестирования» (2 часа)

Разработайте автоматические тесты результата прошлой практической работы. С этой целью: а) выделите вычислительные функции в отдельный класс; б) разработайте UNIT-тесты для данного класса; в) разработайте UI-тест для приложения.

Используйте возможности фреймворка при реализации данной работы

2.11 Практическая работа «MVVM» (4 часа)

Осуществите рефакторинг проекта, полученного в предыдущей практической работе, путём внедрения паттерна MVVM. Обратите внимание на то, что для реализации прослушивания изменений значений во ViewModel (и в Model при необходимости) необходимо использовать Flow.

2.12 Практическая работа «Работа со списками» (4 часа)

Скопируйте часть Model из результата практической работы «Особенности ООП в Kotlin», реализуйте интерфейс с тем же функционалом в Android (Jetpack Compose) с использованием паттерна MVVM.

2.13 Практическая работа «Dependency Injection» (2 часа)

Реализуйте использование Hilt для реализации Dependency Injection в результате прошлой практической работы.

2.14 Практическая работа «Фоновые вычисления» (4 часа)

В данной работе необходимо осуществить с использованием длинной арифметики достаточно долгое вычисление. Программа в ходе выполнения вычисления не должна «зависать». Должна быть возможность остановить вычисление по желанию пользователя.

Подсказка: в работе разрешено использовать BigInteger (не возбраняется реализовать длинную арифметику «руками»).

Вычисление должно осуществляться внутри Service в отдельном потоке, после вычисления результаты должны появиться в Activity, а если он неактивен, то должно появиться оповещение, кликнув по которому будет осуществлен переход на Activity с ответом.

1. Реализуйте программу вычисления $n!$ со всеми десятичными знаками, где $n \in [1 \dots 100000]$
2. Реализуйте программу вычисления 2^n со всеми десятичными знаками, где $n \in [1 \dots 1000000000]$
3. Реализуйте программу вычисления f_n со всеми десятичными знаками, где $n \in [1 \dots 1000000000]$, где f_n – числа Фибоначчи, $f_1 = f_2 = 1$.
4. Реализуйте программу вычисления $n!!$ со всеми десятичными знаками, где $n \in [1 \dots 100000]$
5. Реализуйте программу вычисления f_{2n} со всеми десятичными знаками, где $n \in [1 \dots 1000000000]$, где f_n – числа Фибоначчи, $f_1 = f_2 = 1$.
6. Реализуйте программу вычисления 3^n со всеми десятичными знаками, где $n \in [1 \dots 1000000000]$
7. Реализуйте программу вычисления $f_1 + f_2 + \ldots + f_n$ со всеми десятичными знаками, где $n \in [1 \dots 10000000]$, где f_n – числа Фибоначчи, $f_1 = f_2 = 1$.
8. Реализуйте программу вычисления $1! + 2! + 3! + \ldots + n!$ со всеми десятичными знаками, где $n \in [1 \dots 10000]$
9. Реализуйте программу вычисления $1!! + 2!! + 3!! + \ldots + n!!$ со всеми десятичными знаками, где $n \in [1 \dots 10000]$
10. Реализуйте программу вычисления $f_2 + f_4 + \ldots + f_{2n}$ со всеми десятичными знаками, где $n \in [1 \dots 10000000]$, где f_n – числа Фибоначчи, $f_1 = f_2 = 1$.
11. Реализуйте программу вычисления $2^1 + 2^2 + \ldots + 2^n$ со всеми десятичными знаками, где $n \in [1 \dots 100000000]$
12. Реализуйте программу вычисления $3^1 + 3^2 + \ldots + 3^n$ со всеми десятичными знаками, где $n \in [1 \dots 100000000]$
13. Реализуйте программу вычисления $(n!)!$ со всеми десятичными знаками, где $n \in [1 \dots 9]$, где f_n – числа Фибоначчи, $f_1 = f_2 = 1$.
14. Реализуйте программу вычисления $(n!)!!$ со всеми десятичными знаками, где $n \in [1 \dots 9]$, где f_n – числа Фибоначчи, $f_1 = f_2 = 1$.
15. Реализуйте программу вычисления $(n!!)!$ со всеми десятичными знаками, где $n \in [1 \dots 14]$, где f_n – числа Фибоначчи, $f_1 = f_2 = 1$.
16. Реализуйте программу вычисления $(n!!)!!$ со всеми десятичными знаками, где $n \in [1 \dots 14]$, где f_n – числа Фибоначчи, $f_1 = f_2 = 1$.

17. Реализуйте программу вычисления $f_{n!}$ со всеми десятичными знаками, где $n \in [1 \dots 13]$, где f_n – числа Фибоначчи, $f_1 = f_2 = 1$.

18. Реализуйте программу вычисления $f_{n!!}$ со всеми десятичными знаками, где $n \in [1 \dots 20]$, где f_n – числа Фибоначчи, $f_1 = f_2 = 1$.

19. Реализуйте программу вычисления f_{2^n} со всеми десятичными знаками, где $n \in [1 \dots 30]$, где f_n – числа Фибоначчи, $f_1 = f_2 = 1$.

20. Реализуйте программу вычисления f_{3^n} со всеми десятичными знаками, где $n \in [1 \dots 20]$, где f_n – числа Фибоначчи, $f_1 = f_2 = 1$.

21. Реализуйте программу вычисления $(f_n)!$ со всеми десятичными знаками, где $n \in [1 \dots 21]$, где f_n – числа Фибоначчи, $f_1 = f_2 = 1$.

22. Реализуйте программу вычисления $(f_n)!!$ со всеми десятичными знаками, где $n \in [1 \dots 21]$, где f_n – числа Фибоначчи, $f_1 = f_2 = 1$.

23. Реализуйте программу вычисления f_{f_n} со всеми десятичными знаками, где $n \in [1 \dots 45]$, где f_n – числа Фибоначчи, $f_1 = f_2 = 1$.

24. Реализуйте программу вычисления 2^{f_n} со всеми десятичными знаками, где $n \in [1 \dots 45]$, где f_n – числа Фибоначчи, $f_1 = f_2 = 1$.

25. Реализуйте программу вычисления 3^{f_n} со всеми десятичными знаками, где $n \in [1 \dots 45]$, где f_n – числа Фибоначчи, $f_1 = f_2 = 1$.

2.15 Практическая работа «Content provider, ROOM, фотографии» (6 часов)

В данной работе модифицируется результат практической работы «Dependency Injection» (Android).

1. внедрите сохранение результатов между сеансами работи с использованием Room;
2. внедрите возможность осуществлять прикрепление фотографии к хранимым сущностям, фотографии сохраняйте в файловой системе, а ссылки на них – в базе данных (посредством Room).
3. дайте возможность работать другим приложениям с вашей базой данных (создайте Content Provider);
4. реализуйте протип приложения, работающий с вашим Content Provider.

2.16 Практическая работа «Сенсоры» (4 часа)

Разработайте мобильное приложение, предназначенное для отслеживания активности пользователя согласно варианту.

Информация о сохраненных активностях должна сохраняться между сеансами, пользователь должен иметь возможность корректировать результаты неточных измерений.

Ваша задача повысить точность измерения до возможного максимума.

Обратите внимание на использование всех изученных рекомендованных подходов к проектированию приложения.

Замечание: задание апробируется, потому некоторые варианты могут быть недостаточно корректными.

1. учёта количества и глубины приседаний (в предположении, что телефон в руке, а рука поднимается во время приседа вверх);
2. учёта количества отжиманий (в предположении, что телефон в кармане брюк);
3. учёта количества и высоты прыжков на месте (телефон – в кармане брюк);
4. учёт длины прыжка в длину (телефон – в кармане брюк);
5. учёт длительности выполнения планки (телефон – в кармане брюк, изначально человек стоит, а потом переходит в позу планки, в конце – встаёт);
6. измерение расстояния между точками (пользователь идет в одну точку, нажимает кнопку, идет в другую точку, нажимает кнопку);
7. учёта количества шагов и скорости при беге на месте (в предположении, что телефон в кармане брюк);
8. учёта количества подтягиваний (в предположении, что телефон в кармане брюк);
9. учёта количества подъема туловища из положения лёжа (в предположении, что телефон - в кармане толстовки);
10. учёта количества отжиманий от стены (в предположении, что телефон в кармане толстовки);
11. учёта количества выпадов вперёд (в положении стоя, телефон – в кармане брюк);
12. учёта количества выпадов в сторону (в положении стоя, телефон – в кармане брюк);
13. учёта количества выпадов назад (в положении стоя, телефон – в кармане брюк);
14. учёта количества вращений обруча (телефон – в кармане брюк);
15. учёта количества и качества выполнений виньясы (телефон – в кармане брюк);
16. учёта количества прыжков через скакалку (телефон – в кармане брюк);
17. измерение глубины и прогресса наклона вперёд (пользователь держит телефон в руке);
18. измерение качества выполнения мостика (телефон лежит на животе);
19. измерение высоты вытяжения (телефон поднимается максимально высоко над головой);

20. измерение количества и качества выполнения взмахов рук в противоположные стороны (одна рука вверх, другая вниз, телефон в руке);
21. подсчёт времени проведенном в сидячем положении (телефон в кармане брюк);
22. подсчёт времени пробегания 30 метров;
23. подсчёт количества отжиманий, выполненных за 60 секунд;
24. подсчёт количества приседаний, выполненных за 60 секунд;
25. подсчёт количества подтягиваний, выполненных за 60 секунд.

2.17 Практическая работа «REST API» (6 часов)

Обеспечьте работу с минимум одним rest-api запросом (запрос должен выполняться в фоне), имеющем аргумент, возможности просмотра загруженной информации при отсутствии Интернет-соединения. Реализуйте автоматическое UNIT- и UI-тестирование. UI-тестирование можно осуществлять для классов ROOM (в этом случае Unit-тесты надо перенести в папку AndroidTest). Применяйте изученные архитектурные подходы.

Квалификационный экзамен модуля является вариантом данной практической работы.

1. Разработайте клиент `https://countrylayer.com` с сохранением загруженной информации на мобильном устройстве и загрузкой информации в отдельных потоках. Реализуйте автоматическое UNIT-тестирование и тестирование UI.

   Осуществите поиск стран по языку (для работы используйте бесплатный тарифный план).

2. Разработайте клиент `https://countrylayer.com` с сохранением загруженной информации на мобильном устройстве и загрузкой информации в отдельных потоках. реализуйте автоматическое unit-тестирование и тестирование ui.

   Осуществите поиск стран по валюте (для работы используйте бесплатный тарифный план).

3. Разработайте клиент `https://countrylayer.com` с сохранением загруженной информации на мобильном устройстве и загрузкой информации в отдельных потоках. реализуйте автоматическое unit-тестирование и тестирование ui.

   Осуществите поиск стран по региону (для работы используйте бесплатный тарифный план).

4. Разработайте клиент `https://countrylayer.com` с сохранением загруженной информации на мобильном устройстве и загрузкой информации в отдельных потоках. реализуйте автоматическое unit-тестирование и тестирование ui.

   Осуществите поиск стран по региону (для работы используйте бесплатный тарифный план).

5. Разработайте клиент `https://countrylayer.com` с сохранением загруженной информации на мобильном устройстве и загрузкой информации в отдельных потоках. реализуйте автоматическое unit-тестирование и тестирование ui.

   Осуществите поиск стран по региональному блоку (для работы используйте бесплатный тарифный план).

6. Разработайте клиент `https://github.com/astrocatalogs/OACAPI` с сохранением загруженной информации на мобильном устройстве и загрузкой информации в отдельных потоках. Реализуйте автоматическое UNIT-тестирование и тестирование UI.

   Осуществите поиск всех объектов на данном расстоянии в световых секундах от заданной точки.

7. Разработайте клиент `https://github.com/astrocatalogs/OACAPI` с сохранением загруженной информации на мобильном устройстве и загрузкой информации в отдельных потоках. Реализуйте автоматическое UNIT-тестирование и тестирование UI.

   Осуществите поиск всех объектов на данном расстоянии в световых секундах от заданной точки.

8. Разработайте клиент `https://api.met.no/weatherapi/airqualityforecast/0.1/documentation` с сохранением загруженной информации на мобильном устройстве и загрузкой информации в отдельных потоках.

   Осуществите вывод информации о качестве воздуха по идентификатору станции (достаточно выводить часть информации в RecyclerView – только по выбранным вами характеристикам воздуха).

9. Разработайте клиент `https://api.met.no/weatherapi/locationforecast/2.0/documentation` с сохранением загруженной информации на мобильном устройстве и загрузкой информации в отдельных потоках. Место, по которому отображается информация, можно выбирать из фиксированного списка.

   По данным координатам выведите информацию о прогнозе погоды (можно выводить часть информации – только по выбранным вами характеристикам погоды).

10. Разработайте клиент `http://www.tvmaze.com/api` с сохранением загруженной информации на мобильном устройстве и загрузкой информации в отдельных потоках.

    По данной поисковой строке найдите информацию о шоу.

11. Разработайте клиент `http://www.tvmaze.com/api` с сохранением загруженной информации на мобильном устройстве и загрузкой информации в отдельных потоках.

    По данной поисковой строке найдите информацию о героях шоу.

12. Разработайте клиент `http://www.tvmaze.com/api` с сохранением загруженной информации на мобильном устройстве и загрузкой информации в отдельных потоках.

    По данной дате и стране найдите информацию о расписании шоу (SCHEDULE).

13. Разработайте клиент `http://www.tvmaze.com/api` с сохранением загруженной информации на мобильном устройстве и загрузкой информации в отдельных потоках.

    По данной дате и стране найдите информацию о расписании шоу в стримах (SCHEDULE/WEB).

14. Разработайте клиент `https://date.nager.at/Api` с сохранением загруженной информации на мобильном устройстве и загрузкой информации в отдельных потоках.

    По данному году и коду страны найдите информацию о выходных днях.

15. Разработайте клиент `https://newton.now.sh/` с сохранением загруженной информации на мобильном устройстве и загрузкой информации в отдельных потоках.

    По данному выражению найдите его упрощенный вариант. Все результаты сохраняйте и выводите в списке (начальное выражение – результат).

16. Разработайте клиент `https://docs.tronalddump.io/` с сохранением загруженной информации на мобильном устройстве и загрузкой информации в отдельных потоках.

    По данному фрагменту текста найдите подходящие цитаты. Для указания header можно использовать @Header перед описанием функции.

17. Разработайте клиент `https://alexwohlbruck.github.io/cat-facts/` с сохранением загруженной информации на мобильном устройстве и загрузкой информации в отдельных потоках.

    По данному классу животных найдите факты о них.

18. Разработайте клиент `https://github.com/RocktimSaikia/anime-chan` с сохранением загруженной информации на мобильном устройстве и загрузкой информации в отдельных потоках.

    По данному аниме выведите цитаты из него.

19. Разработайте клиент `https://github.com/RocktimSaikia/anime-chan` с сохранением загруженной информации на мобильном устройстве и загрузкой информации в отдельных потоках.

    По данному персонажу выведите его цитаты.

20. Разработайте клиент `https://openlibrary.org` с сохранением загруженной информации на мобильном устройстве и загрузкой информации в отдельных потоках.

    По данной теме выведите информацию о книгах (не обязательно всех).

21. Разработайте клиент `https://openlibrary.org` с сохранением загруженной информации на мобильном устройстве и загрузкой информации в отдельных потоках.

    По данному автору выведите информацию о произведениях.

22. Разработайте клиент `https://openlibrary.org` с сохранением загруженной информации на мобильном устройстве и загрузкой информации в отдельных потоках.

    По данному фрагменту названия выведите информацию о произведениях.

23. Разработайте клиент `https://github.com/thundercomb/poetrydb#readme` с сохранением загруженной информации на мобильном устройстве и загрузкой информации в отдельных потоках.

    По данному авторы выведите его произведения (начальные фрагменты).

24. Разработайте клиент `https://github.com/fawazahmed0/currency-api#readme` с сохранением загруженной информации на мобильном устройстве и загрузкой информации в отдельных потоках.

    По данной валюте выведите обменные курсы с ней.

25. Разработайте клиент `https://https://nationalize.io/` с сохранением загруженной информации на мобильном устройстве и загрузкой информации в отдельных потоках.

    По данному имени выводите гипотетические страны и вероятности совпадения страны.

26. Разработайте клиент `https://www.boredapi.com` с сохранением загруженной информации на мобильном устройстве и загрузкой информации в отдельных потоках.

    По данному типу и количеству участников выведите подходящее занятие. Предыдущие запросы и результаты сохраняйте и выводите в RecyclerView.

27. Разработайте клиент `https://github.com/davemachado/public-api` с сохранением загруженной информации на мобильном устройстве и загрузкой информации в отдельных потоках.

    По данному фрагменту названия выведите информацию о публичных сервисах.

28. Разработайте клиент `https://newton.now.sh/` с сохранением загруженной информации на мобильном устройстве и загрузкой информации в отдельных потоках.

    По данному выражению найдите его разложение на множители. Все результаты сохраняйте и выводите в списке (начальное выражение – результат).

29. Разработайте клиент `https://newton.now.sh/` с сохранением загруженной информации на мобильном устройстве и загрузкой информации в отдельных потоках.

    По данному выражению найдите его производную. Все результаты сохраняйте и выводите в списке (начальное выражение – результат).

30. Разработайте клиент `https://newton.now.sh/` с сохранением загруженной информации на мобильном устройстве и загрузкой информации в отдельных потоках.

    По данному выражению найдите его интеграл. Все результаты сохраняйте и выводите в списке (начальное выражение – результат).

2.18 Последнее занятие (2 часа)

Последнее занятие предназначено для сдачи работ, которые не были приняты заранее. На каждого студента выделяется равное время, поэтому сдать все работы в последний день не удастся.

3 Список литературы

1. Официальная документация по языку Kotlin: `https://kotlinlang.org/docs/reference/`
2. Официальная документация по платформе Android: `https://developer.android.com/docs`
3. Быстрое введение в Kotlin от авторов языка: `https://stepik.org/course/4222/promo`