Сборник заданий для семинарских занятий
по курсу
«Разработка мобильных приложений»

Содержание

1 Общие сведения **3**

2 Язык Kotlin **3**
2.1 Семинар «Разработка первой программы на Kotlin» (2 часа) 4
2.2 Семинар «Условные конструкции» (2 часа) 6
2.3 Семинар «Массивы в Kotlin» (2 часа) 7
2.4 Семинар «Строки в Kotlin» (2 часа) 10
2.5 Практическая работа «функции Kotlin» (2 часа) 12
2.6 Семинары «Функции высшего порядка в Kotlin/Функциональный подход в Kotlin» (4 часа) 16
2.7 Семинар «Поддержка ООП в Kotlin» (2 часа) 20
2.8 Семинар «Перегрузка операторов в Kotlin» (2 часа) 23
2.9 Семинар «Делегирование в Kotlin» (2 часа) 24
2.10 Семинар «Абстрактное программирование в Kotlin» 27

3 Разработка на языке Kotlin для платформы Android **29**
3.1 Семинар «Создание мобильного приложения» (2 часа) 30
3.2 Семинар «Простейшие программы для Android» (2 часа) 31
3.3 Семинары «Сервисы/Уведомления/Работа с другими приложениями» (6 часов) 33
3.4 Семинар «Работа с базами данных/навигация» (4 часа) 35
3.5 Семинар «Работа с внешними сервисами RestAPI» (2 часа) 37
3.6 Семинар «Автоматизация тестирования» (2 часа) 41
3.7 Семинар «Проектирование приложения с использованием MVVM» (2 часа) . 42
3.8 Семинар «Паттерн репозиторий» (2 часа) 43
3.9 Семинар «Использование библиотек для поддержки Dependency Injection» (2 часа) 44
3.10 Последний семинар 45

4 Список литературы **45**

1 Общие сведения

Сборник содержит задания для семинарских занятий по курсу «Разработка мобильных приложений».

Задачник расчитан на 48 часов семинарских занятий.

Структурно курс делится на две части:

1. изучение языка Kotlin,

2. изучение разработки на платформе Android.

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 Семинар «Разработка первой программы на Kotlin» (2 часа)

Цель первых четырех семинаров – обеспечить подготовку к выполнению заданий, нацеленных на освоение особенностей языка Kotlin. В первой работе необходимо написать небольшую программу, чтобы убедиться в понимании базовых конструкций языка (функция main, ветвления, циклы) и целый тип, 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).

Задание

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

2.2 Семинар «Условные конструкции» (2 часа)

Во второй работе предполагается освоение оператора when.

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 задачах предполагается, что не используются специальные функции, предназначенные для работы с датами.

1. Даны год, месяц и число (в виде чисел) – выведите год, месяц и число для следующего дня (месяц надо выводить в виде названия, например «1 января 2000 года»).

2. Даны год, месяц и число (в виде чисел) – выведите год, месяц и число для предыдущего дня (месяц надо выводить в виде названия, например «1 января 2000 года»).

3. Дано число от 1 до 1000 – выведите его на русском языке, например «семьсот семьдесят семь».

4. Дано число от 1 до 1000 – выведите его на английском языке, например «seven hundred seventy-seven».

5. Даны год, месяц и число (в виде чисел) – выведите год, месяц и число для дня, который будет через неделю (месяц надо выводить в виде названия, например «1 января 2000 года»).

6. Даны год, месяц и число (в виде чисел) – выведите год, месяц и число для дня, который был неделю назад (месяц надо выводить в виде названия, например «1 января 2000 года»).

7. Даны две даты одного года (то есть 5 чисел – год, месяц, число, месяц, число) – выведите фразу «Между указанными датами прошло *столько-то (словами)* дней».

8. По году, месяцу и числу выведите знак зодиака.

9. Даны год, месяц и число (в виде чисел) – выведите год, месяц и число для того же дня следующего месяца, а если его не существует, то первого дня месяца, что идет после следующего (месяц надо выводить в виде названия, например «1 января 2000 года»).

10. Даны год, месяц и число (в виде чисел) – выведите год, месяц и число для того же дня предыдущего месяца, а если его не существует, то последнего дня месяца, что идет до предыдущего (месяц надо выводить в виде названия, например «1 января 2000 года»).

11. Даны две даты одного года (то есть 5 чисел – год, месяц, число, месяц, число) – выведите фразу «*Number (for exampe, seventy-seven)* years have passed between the dates indicated» на английском языке.

2.3 Семинар «Массивы в Kotlin» (2 часа)

В третьей работе предполагается освоение массивов языка Kotlin.

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).

Во всех заданиях предполагается использование массивов (не списков, не множеств и других подобных структур), которые оптимальны по скорости для больших по объему исходных данных

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.

2.4 Семинар «Строки в Kotlin» (2 часа)

В последней ознакомительной работе предполагается освоение работы со строками языка Kotlin (без использования специальных функций).

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 симвлов 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 символов 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).

2.5 Практическая работа «функции Kotlin» (2 часа)

Задание №1 (функции и лямбда-функции)

Разработайте программу в соответствии с вашим заданием при этом:

- основной алгоритм без ввода-вывода оформите в виде функции
- в функцию передавайте как исходное число, так и условие отбора в виде лямбда-функции (проверка на четность/нечетность/кратность трем/некратность трем) в зависимости от варианта; функция должна корректно работать при изменении условия отбора

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 (функции высшего порядка)

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 $f(x)$, $f(f(x))$, $f(f(f(x)))$, ...

2.6 Семинары «Функции высшего порядка в Kotlin/Функциональный подход в Kotlin» (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 первой, третьей и четвертой практических работ.

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.

2.7 Семинар «Поддержка ООП в Kotlin» (2 часа)

Напишите программу осуществляющую ввод информации о сущностях, описанных в вашем варианте задания и вывод на экран некоторых из них. Количество вводимых сущностей не ограничено; обязательно использовать ООП, инкапсуляцию, наследование и полиморфизм. Проверять корректность входных данных и делать проверку того, что хватает памяти не обязательно.

1. Товары Интернет-магазина – книги и диски. Все товары определяются ценой, книги имеют название, автора, количество страниц; диски – название, количество треков. Выведите на экран все товары со стоимостью меньше 100 рублей.

2. Преподаватели определяются ФИО. Для тех, кто имеют диссертацию дополнительно вводится ее название; для остальных – стаж работы. Вывести всех преподавателей, у которых ФИО начинается на букву «А».

3. Телефоны определяются названием модели. Проводные телефоны дополнительно определяются типом номеронабирателя (диск или кнопки); а беспроводные – дальностью действия радиосигнала. Вывести все телефоны, название которых начинается на «А».

4. Покатушки определяются названием и расстоянием. Однодневные катушки дополнительно определяются плановым временем поездки (в часах). Многодневные катушки определяются количеством дней и категорией сложности похода (от 1 до 6). Вывести все покатушки длиной более 100 км.

5. Музыкальная композиция определяется названием и композитором. Дополнительно для песни указывается автор стихов. Выведите информацию о всех композициях, у которых композитор начинается на букву «А».

6. Олимпиада определяется названием. Если олимпиада участвует в программе приема в ВУЗы дополнительно указывается уровень олимпиады (1–3), если олимпиада – этап всероссийской, то указывается название этапа (школьная, окружная, региональная, всероссийская), в остальных случаях – размер призового фонда. Выведите все олимпиады, название которых начинается на букву «А».

7. Проездной билет определяется стоимостью. Билет на количество поездок определяется количеством поездок. Билет на неограниченное количество поездок определяется сроком действия (1 день, 5 дней, 10 дней, 15 дней, месяц, три месяца, 6 месяцев, год). Выведите информацию о билетах, стоимостью меньше 300 рублей.

8. Информация о студенте определяется ФИО. Для студентов, не имеющих автомата, указывается балл, полученный на экзамене (2–5); для студентов, имеющих автомат указывается основание (олимпиада или контрольные работы). В случае, если контрольная работа – то также указывается средний балл за к/р. Выведите всю информацию о студентах с фамилией, начинающейся на буквы от А до К.

9. Сотовый телефон определяется названием. Для смартфонов указывается операционная система. А для других телефонов – наличие браузера. Выведите информацию о телефонах, название которых содержит слово «Nokia».

10. Куртка определяется названием модели, наличием капюшона. Для мембранных курток указывается степень водонепроницаемости (число в мм рт. ст.), для остальных – наличием пропитки. Выведите информацию обо всех куртках, имеющих капюшон.

11. Жесткий диск определяется названием и емкостью. Внешние жесткие диски определяются дополнительно наличием системы, смягчающей последствия падения. Внутренние жесткие диски – размером (2.5/3.5 дюйма). Выведите информацию о дисках, емкостью больше 200 Гб.

12. Велосипед определяется названием модели. Горному велосипеду соответствует количество скоростей, BMX – тип конструкции (фривил, кассетная, фрикостер). Выведите информацию обо всех велосипедах, содержащих в названии «Norco».

13. Электронная книга определяется названием и размером экрана. Для EInk-дисплея указывается поколение (pearl, vizplex); для LCD – количество поддерживаемых цветов. Выведите информацию о всех книгах с размером экрана не менее 7 дюймов.

14. GPS определяется названием, диагональю экрана. Для переносных GPS указывается наличие велосипедного крепления; для автомобильных – поддержка отображения пробок и наличие радар-детектора. Выведите информацию обо всех GPS с размером экрана менее 7 дюймов.

15. Пылесос определяется названием модели. Для обычного пылесоса указывается мощность, для пылесоса-робота – размер убираемого помещения и количество виртуальных стен. Выведите информацию обо всех пылесосах, содержащих в названии слово Indesit.

16. Туры определяются названием. Для пляжного тура указывается тип пляжа (галечный, песок); для экскурсионного – количеством экскурсий. Выведите информацию обо всех турах, содержащих слово Египет.

17. Язык программирования определяется названием. Алгоритмические языки определяются поддержкой ООП (отсутствует, на классах, прототипная), остальные языки – типом (функциональный, логический, стиль). Выведите информацию обо всех языках, название которых начинается с буквы «А».

18. Контрагенты определяются названием. Индивидуальные предприниматели дополнительно определяются наличием счета в банка, а юридические лица – формой организации (ООО, ОАО. ЗАО). Выведите информацию обо всех контрагентах, название которых начинается с буквы «А».

19. Счет в банке определяется номером. Для текущего счета указывается плата за обслуживание, для сберегательного счета – проценты годовых и наличие капитализации. Выведите информацию обо всех счетах, номер которого начинается с 408178...

20. Автомобильная дорога определяется названием и километражом. Бесплатная дорога определяется статусом автомагистрали (автомагистраль или нет), а платная – стоимостью за километр для обычных пользователей. Выведите информацию о дорогах, длина которых менее 100 км.

21. Офисное здание определяется адресом. В случае наличия стоянки указывается количество машиномест и стоимость аренды за месяц. Выведите информацию о зданиях, в адресе которых присуствует слово Тверская.

22. Товары Интернет-магазина – GPS-навигаторы и карты. Все товары определяются ценой и названием, GPS-навигаторы имеют назначение (ручной, автомобильный) и признак возможности загрузки карт; карты – размером (в Мб). Выведите информацию о всех товарах со стоимостью менее 4000 рублей.

23. Товары Интернет-магазина – чаи и кофе. Все товары определяются ценой, названием и весом, кофе – типом (растворимый, молотый, в зернах), чаи – типом (черный, зеленый). Выведите информацию о всех товарах с весом менее 150 г.

24. Объекты продаваемые в коттеджном поселке: участки (определяются площадью, стоимостью, наличием подряда), дома (определяются этажностью, площадью и стоимостью). Выведите все объекты со стоимостью меньше 1000000 рублей.

25. Вопросам теста соответствует формулировка и количество баллов за правильный ответ. Вопросам с вариантами правильных ответов соответствует 4 варианта ответа и номер правильного ответа; остальным вопросам – формулировка правильного ответа. Выведите все вопросы, оцениваемые в 10 баллов и выше.

26. Слова определяются собственно словом. Для существительных указывается род, для глаголов – спряжение. Выведите информацию обо всех словах, начинающихся на букву «А».

27. Операционная система определяется названием. Для операционной системы на базе Linux указывается название менеджера пакетов; для остальных – стоимость лицензии. Вывести все операционные системы, у которых название начинается на букву «А».

28. Рюкзаки определяются названием модели и емкостью. Для городских рюкзаков указывается наличие «вентилируемой спины» для походных – количество отделений и наличие крепления для трекинговых палок. Вывести информацию обо всех рюкзаках, в названии которых присутствует слово «Trek».

29. Автостоянка определяется названием, количеством машиномест. Для крытой автостоянки указывается количество этажей. Для открытой стоянки – наличие охраны. Вывести информацию обо всех автостоянках с количеством мест больше 20.

30. Партия определяется названием. Для тех партий, что финансируются из бюджета указывается размер ассигнований, а для остальных – количество депутатов в каких-либо представительных органах власти. Вывести информацию обо всех партиях с названиями, начинающимися на буквы от «А» до «К».

2.8 Семинар «Перегрузка операторов в Kotlin» (2 часа)

1. Реализуйте класс комплексного числа с реализацией операций +,-,*,/
2. Реализуйте класс обыкновенной дроби (состоящей из целой части, числителя и знаменателя) с реализацией операций +,-,*,/
3. Реализуйте класс 3-х мерного вектора с реализацией операторов +,-,* (векторное произведение), % (скалярное произведение)
4. Реализуйте класс матрицы 2×2 с реализацие операторов +,-,*, /.

2.9 Семинар «Делегирование в Kotlin» (2 часа)

Реализуйте несколько классов, предоставляющих возможность реализации паттерна Decorator посредством делегирования языка Kotlin. Обратите внимание на то, что представленные задания невозможно решить с помощью наследования без дублирования кода или изобретения сложных механизмов (кроме языков с множественным наследованием).

1. Осуществить ввод информацию о преподавателях и вывод информации о тех из них, у кого средний балл учащихся выше 4. Преподаватели характеризуются фамилией, именем, отчеством, полом, средним балом учащихся. Если преподаватель имеет категорию, то он также характеризуется датой последнего подтверждения категории и номером приказа об этом. Если преподаватель имеет кандидатскую степень, то он также характеризуется темой диссертации, научным направлением и датой защиты.

2. Осуществить ввод информации о студентах и вывод информации о тех из них, у кого средний балл выше 4. Студент характеризуется фамилией, именем, отчеством, полом, средним баллом. Если студент – платник, то также номером договора и признак отсутствия задолженности по оплате. Если студент – военнообязанный, то также номером приписного свидетельства или военного билета, названием военкомата, признаком того, что он проходил воинскую службу.

3. Осуществить ввод информации об остановочных пунктах пригородного железнодорожного транспорта. Вывести информацию о тех, где пассажиропоток выше 1000 человек/сутки. Остановочный пункт характеризуется названием, пассажиропотоком, расстоянием от вокзала. Если остановочный пункт имеет турникетный комплекс, то он также характеризуется количеством турникетных павильонов, стоимостью билета «на выход». Если остановочный пункт расположен вблизи станции метро, то он характеризуется названием ближайшей станции метро и расстоянием до нее.

4. Осуществите ввод информации о ВУЗах и вывод информации о тех из них, где учится больше 2000 человек. ВУЗ характеризуется названием, количеством студентом, адресом, номером лицензии. Если ВУЗ имеет аккредитацию, то он дополнительно характеризуется номером аккредитации и датой окончания аккредитации. Если ВУЗ выдает документы международного образца то он дополнительно характеризуется стоимостью выдачи такого документа и регионом, где он признается (например, весь мир, ЕС, США).

5. Осуществите ввод информации о планшетах и вывод информации о тех из них, что стоят меньше 15000 рублей. планшет характеризуется названием, производителем, стоимостью. Если планшет поддерживает использование SIM-карты то дополнительно указывается поддержка 3G, поддержка LTE. Если планшет поддерживает Bluetooth, то указывается какой именно стандарт Bluetooth поддерживается, возможно ли подключить Bluetooth клавиатуру.

6. Осуществить ввод информации о смартфонах и вывод информации о тех из них, что стоят меньше 10000 рублей. Смартфон определяется названием модели, названием производителя, ценой. Если смартфон поддерживает WiFi, то дополнительно указывается стандарт WiFi и максимальная стоимость. Если смартфон поддерживает геопозиционирование, то указывается поддерживает ли он GPS, поддерживает ли он глонасс и количество каналов.

7. Осуществить ввод информации о телевизорах и вывод о тех из них, что стоят меньше 10000 рублей. Телевизор определяется названием марки, названием производителя, размером диагонали. Если телевизор поддерживает Smart TV, то дополнительно указывается название стандрта Smart TV, предварительно установленные программы. Если телевизор поддерживает многоканальный звук то дополнительно указываетс наличие оптического выхода, наличие поканальных выходов,

8. Осуществить ввод информации о накопителях и вывод о тех из них, у которых емкость меньше 100 Гб. Накопитель определяется названием модели, названием производителя, емкостью в гигабайтах. Если накопитель имеет интерфейс USB, то дополнительно указывается номер стандарта USB и скорость передачи информации. Если накопитель – жесткий диск, то дополнительно указывается количество поверхностей и размер (физический).

9. Осуществить ввод информации о компьютерах и вывести информацию о компьютерах со стоимостью меньше 20000 рублей. Компьютер определяется названием производителя, названием модели, стоимостью. Если компьютер является моноблоком, то он дополнительно определяется размером диагонали дисплея. Если компьютер поддерживается WiFi, то дополнительно указывается стандарт WiFi и максимальная поддерживаемая скорость.

10. Осуществить ввод информации о телеканалах и вывести информацию о тех из них, аудитория которых превышает 1000000. Телеканал определяется названием и размером аудитории. Если телеканал является эфирным, то дополнительно указывается диапазон (МВ, ДМВ) и частота. Если телеканал является государственным, то дополнительно указывается размер его финансирования.

11. Осуществить ввод информации о реках и выведите на экран те из них, что имеют длину более 20 км. Река определяется названием и длиной. Если река является судоходной, то дополнительно указывается ширина и глубина основной части реки. Если река является водоснабжающей, то указывается суточный водозабор и название водопроводной станции.

12. Осуществить ввод информации о памятниках и вывести информацию о тех из них, у которых рейтинг выше 2.5. памятник определяется названием, адресом и рейтингом. Если памятник посвящен одному человеку, то указывается ФИО этого человека и годы его жизни. Если памятник связан с военными действиями, то указывается название войны и годы войны.

13. Осуществить ввод информации о ПИФах и вывести информацию о тех, из них, что основаны до 2008 года. ПИФ определяется названием управляющей компании, названием ПИФа и годом основания. Для интервального ПИФа дополнительно указывается начало и окончание следующего интервала (год, месяц, число). Для ПИФа акций указывается акции какого эшелона покупаются (первого, второго, другие).

14. Осуществить ввод информации о вкладах и вывести информацию о тех, у которых доходность больше 9 процентов. Вклад определяется названием, банком и процентом. Для вкладов, где разрешены взносы, дополнительно указывается за сколько дней до окончания можно делать взносы и минимальный размер взноса. Для вкладов, где разрешено снятие, дополнительно указывается через сколько дней после начала вклада разрешено снятие и размер неснижаемого остатка.

15. Осуществить ввод информации о маршрутах электричек и вывести те из них, длина которых превышает 50 км. Маршрут определяется названиями отправного и конечного пункта и длиной маршрута. Если маршрут является экспрессным, то дополнительно указывается стоимость проезда и время в пути. Если маршрут является межобластным, то дополнительно указываются названия областей.

16. Осуществить ввод информации о холодильниках и вывести информацию о тех из них, высота которых превышает 1.5 метра. Холодильник определяется высотой, шириной, глубиной, названиями модели и производителя. Для холодильников с морозильником дополнительно указывается высота, глубина и ширина морозильной камеры. Для холодильников имеющих зону свежести также указывается высота, глубина и ширина зоны свежести.

17. Осуществить ввод информации о наушниках и вывести информацию о тех из них, которые стоят больше 500 рублей. Наушник определяется названием производителя, названием модели, наличием системы активного шумоподавления, ценой. Для беспроводных наушников дополнительно указывается радиус действия и флаг того, что сигнал цифровой. Для вставных наушников дополнительно указывается количество пар сменных амбушюров в комплекте и наличие регулятора громкости.

18. Осуществить ввод информации об электронных книгах и вывести информацию о тех из них, которые стоят больше 3000 рублей. Электронная книга определяется названием модели и названием производителя, стоимостью. Если электронная книга имеет дисплей на основе электронных чернил, то дополнительно указывается цветной дисплей или черно-белый, а также наличие встроенной подсветки. Если электронная книга имеет сенсорный интерфейс, то дополнительно указывается его тип (емкостный, резистивный, индуктивный, инфракрасный).

19. Осуществить ввод информации об устройствах GPS и вывести информацию о тех из них, что стоят больше 5000 рублей. Устройство определяется названиями моделей и производителя, стоимостью и флагом поддержки Глонасс. Если устройство туристическое, то дополнительно указывается флаг возможности нахождения под водой, наличие магнитного компаса и тип батареи (обычная или специальная). Если устройство имеет цветной дисплей, то указывается количество поддерживаемых цветов.

20. Осуществить ввод информации о туристических палатках и вывести информацию о тех из них, у которых водостойкость дна больше 7000. Палатка определяется названием производителя и названием модели, водостойкостью тента и дна. Если палатка имеет внутреннюю палатку, то указывается указывается ширина и высота внутренней палатки. Если палатка многоместная, то указывается количество мест и количество входов.

21. Осуществить ввод информации об обогревателях и вывести информацию о тех из них, что стоят больше 1500 рублей. Обогреватель определяется названиями производителя и модели, стоимостью, мощностью Если обогреватель инфракрасный, то дополнительно указывается тип (галогенный, карбоновый или кварцевый) и флаг наличия автоматического вращения. Если обогреватель режим программирование то указываются флаги возможности отсрочки старта и регулировки температуры.

2.10 Семинар «Абстрактное программирование в Kotlin»

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.

3 Разработка на языке Kotlin для платформы Android

3.1 Семинар «Создание мобильного приложения» (2 часа)

Повторите действия с мастер-класса, проведённого на паре:

1. осуществите инсталляцию Android Studio (при необходимости);
2. создайте проект;
3. разработайте программу, осуществляющую сложение двух чисел;
4. реализуйте корректную поддержку интернационализации и смены конфигурации;
5. модифицируйте программу таким образом, чтобы ответ выводился на втором Activity.

3.2 Семинар «Простейшие программы для Android» (2 часа)

Разработайте программу, работающую под управлением Android с использованием Jetpack Compose. 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.

3.3 Семинары «Сервисы/Уведомления/Работа с другими приложениями» (6 часов)

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.

После получения результата должна быть возможность его отправить в другое приложение.

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, $f_1 = f_2 = 1$.

3.4 Семинар «Работа с базами данных/навигация» (4 часа)

Создайте приложение, позволяющее осуществлять CRUD-операции с использованием пользовательского интерфейса. Постоянное хранение информации осуществляйте с использованием библиотеки Room. Переход между «экранами» осуществляется с использованием библиотеки Navigation (предназначенной для использования с Jetpack Compose).

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.

3.5 Семинар «Работа с внешними сервисами RestAPI» (2 часа)

Обеспечьте работу с минимум одним rest-api запросом (запрос должен выполняться в фоне), имеющем аргумент, возможности просмотра загруженной информации при отсутствии Интернет-соединения. Работа должна осуществляться с использованием библиотеки retrofit.

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).

3.6 Семинар «Автоматизация тестирования» (2 часа)

Разработайте автоматические тесты программы, разработанной на семинаре «Простые мобильные приложения». 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

3.7 Семинар «Проектирование приложения с использованием MVVM» (2 часа)

Осуществите рефакторинг проекта, полученного на предыдущем семинаре, путём внедрения паттерна MVVM. Обратите внимание на то, что для реализации прослушивания изменений значений во ViewModel (и в Model при необходимости) необходимо использовать Flow. Используйте библиотеки Jetpack везде, где они применимы.

3.8 Семинар «Паттерн репозиторий» (2 часа)

Осуществите рефакторинг проекта, полученного на прошлом семинаре, посредством использования паттерна репозиторий.

3.9 Семинар «Использование библиотек для поддержки Dependency Injection» (2 часа)

Используйте Hilt для внедрения Dependency Injection в проект, созданный на прошлом семинаре.

3.10 Последний семинар

Последнее занятие предназначено для подведения итогов, осуществления review кода последнего проекта, который, в идеале, включает корректное использование паттерна MVVM, репозитория и Dependency Injection в проекте, использующем Rest API и кеширование результатов на стороне клиента.

4 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`