



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(ДГТУ)**

Факультет
Информатика и вычислительная техника

Кафедра
Программное обеспечение вычислительной техники и автоматизированных систем

**БАЗОВЫЕ ТИПЫ ДАННЫХ,
УПРАВЛЯЮЩИЕ КОНСТРУКЦИИ И ФУНКЦИИ**

Практикум
по выполнению лабораторной работы №1
по дисциплине «Языки программирования высокого уровня»

Технология разработки сложных программных систем
(указывается направленность (профиль) образовательной программы)

09.04.04 Программная инженерия
(указывается код и наименование направления подготовки)

Ростов-на-Дону
2024 г.

Составители: канд. техн. наук, зав. каф. В.В. Долгов

УДК 004.432

Базовые типы данных, управляющие конструкции и функции: метод. указания. – Ростов н/Д: Издательский центр ДГТУ, 2024.

В методическом указании рассматриваются вопросы создания консольных приложений на языке программирования Kotlin, консольный ввод/вывод, работа с базовыми типами данных и основными управляющими конструкциями языка, а также описание и вызов функций. Дано понятие, назначение и общие вопросы использования исключений как способа обработки ошибочных ситуаций, возникающих при работе программ. Приведены задания к лабораторной работе, помогающие закрепить на практике полученные знания, и контрольные вопросы для самопроверки.

Предназначено для обучающихся по направлению 09.04.04 «Программная инженерия», профиль «Технология разработки сложных программных систем».

Ответственный за выпуск:

зав. кафедрой «Программное обеспечение вычислительной техники и автоматизированных систем» Долгов В.В.

© В.В. Долгов, 2024

© Издательский центр ДГТУ, 2024

1 Теоретическая часть

1.1 Создание консольных приложений

Для создания нового проекта на языке программирования Kotlin [1] в среде IntelliJ IDEA Community Edition [2] необходимо выбрать соответствующий пункт главного меню и в появившемся окне указать наименование проекта, систему сборки на основе самой среды IntelliJ IDEA и установить переключатель (чек-бокс) «Add sample code». В этом случае сразу поле создания такого проекта и его запуска система будет создавать текстовое консольное окно и выводить туда текст «Hello World!». Для ввода/вывода в таком окне используются функции, описанные ниже в таблице 1.

Стоит обратить особое внимание, что в языке отсутствуют средства непосредственного ввода чисел, а весь ввод рассматривается как строковая информация. Для преобразования строковых данных в тип, необходимый программе обычно используют методы семейства «to*», где вместо символа «*» подставляется имя необходимого базового типа данных.

Таблица 1. Основные при работе с консолью

Функция	Описание
print	Выводит переданный в параметрах объект в виде строкового представления на консоль.
println	Выводит переданный в параметрах объект в виде строкового представления на консоль и выполняет перевод строки (следующий вывод будет начинаться с новой строки).
readln	Читает и возвращает текстовую строку из консоли (стандартного входного потока). Если на момент вызова данной функции достигнут конец потока (символ EOF), генерирует исключение ReadAfterEOFException.
readLine	Читает и возвращает текстовую строку из консоли (стандартного входного потока). Если на момент вызова данной функции достигнут конец потока (символ EOF), возвращает null.
readlnOrNull	Синоним функции readLine.

2 Задание к лабораторной работе

Создать в одной из сред программирования (на выбор предлагается IntelliJ IDEA Community Edition или Visual Studio Code [3]) проект консольного приложения на языке Kotlin. Используя функции ввода/вывода информации, реализовать программу в соответствии с вариантом задания (табл. 2). В процессе выполнения работы запрещено использовать стандартные функции или библиотеки языка программирования Kotlin либо среды выполнения JVM, выполняющие значимую часть функционала задания.

Код программы должен содержать обработку ошибочных ситуаций, которые могут возникнуть в ходе выполнения программы. В случае возникновения ошибок, организовать информативный вывод данных о возникшей ошибке на экран с предложением продолжить выполнение, проигнорировав ошибку, или завершить выполнение программы.

При защите работы студент должен уметь: создавать консольные проекты, устанавливать точки останова для отладки программы, выполнять программу пошагово в режиме отладки, просматривать значения переменных при отладке.

Таблица 2. Варианты заданий

№ варианта	Задание к лабораторной работе
1	Разработайте программу для обработки текстовых документов. Программа должна содержать как минимум следующие функции: 1. Функция чтения данных: Читает текст из файла и сохраняет его в строковую переменную. 2. Функция анализа текста: Подсчитывает количество предложений, слов и символов в тексте. 3. Функция поиска и замены: Ищет заданное слово в тексте и заменяет его другим словом.
2	Создайте программу, которая выполняет статистический анализ набора числовых данных. Программа должна содержать как минимум следующие функции: 1. Функция ввода данных: Принимать от пользователя последовательность чисел и сохранять её в массив. 2. Функция вычисления статистики: Вычислять среднее значение, медиану, дисперсию и стандартное отклонение введённых чисел. 3. Функция вывода результатов: Отображать полученные статистические показатели в удобочитаемом формате.
3	Разработайте программу для выполнения операций над матрицами. Программа должна содержать как минимум следующие функции: 1. Функция создания матрицы: Создаёт матрицу заданного размера с вводом элементов пользователем. Допускается чтение матрицы из текстового файла. 2. Функция умножения двух матриц. 3. Функция вывода матрицы: Отображает матрицу в табличном виде на экране.
4	Разработайте программу, которая конвертирует числа между системами счисления. Программа должна содержать как минимум следующие функции: 1. Функция ввода данных: число и основания исходной и целевой систем счисления. 2. Функция преобразования: Переводит число из одной системы счисления в другую. 3. Функция вывода данных: Показывает результат преобразования пользователю.
5	Напишите программу, которая позволяет играть в "Крестики-нолики" с компьюте-

	<p>ром. Программа должна содержать как минимум следующие функции:</p> <ol style="list-style-type: none"> 1. Функция отображения поля: Рисует текущее состояние игрового поля. 2. Функция хода игрока: Позволяет игроку сделать ход, проверяя корректность ввода. 3. Функция хода компьютера: Реализует простой алгоритм выбора хода компьютера.
6	<p>Создайте программу для обработки информации о студентах и их оценках. Программа должна содержать как минимум следующие функции:</p> <ol style="list-style-type: none"> 1. Функция чтения данных: Загружает информацию о студентах и их оценках из файла. 2. Функция вычисления средних баллов: Рассчитывает средний балл для каждого студента и общий средний балл. 3. Функция формирования отчёта: Генерирует отчёт с рейтингом студентов по среднему баллу.
7	<p>Разработайте программу для работы с геометрическими фигурами на координатной плоскости. Программа должна:</p> <ol style="list-style-type: none"> 1. Функция ввода координат: Принимает от пользователя координаты вершин многоугольника. 2. Функция вычисления периметра и площади: Рассчитывает периметр и площадь заданного многоугольника. 3. Функция обнаружения пересечений: Вводит координаты двух многоугольников и проверяет, пересекаются ли они на плоскости. Если многоугольники пересекаются, возвращает площадь области пересечения. Для простоты считать, что все многоугольники выпуклые.
8	<p>Разработайте программу для генерации и проверки уникальных паролей. Программа должна содержать как минимум следующие функции:</p> <ol style="list-style-type: none"> 1. Функция генерации случайного пароля заданной длины, содержащего символы из заданного алфавита. 2. Функция проверки надежности пароля (наличие букв верхнего и нижнего регистра, цифр и специальных символов). 3. Функция, которая генерирует указанное количество уникальных паролей заданной длины.

3 Материально-техническое обеспечение работы

Аудитория для проведения лабораторных занятий должна быть укомплектована специализированной мебелью и индивидуальными компьютерами следующей минимальной комплектации:

- Процессор: не менее двух исполнительных ядер, совместимый с системой команд x86 и x64, с поддержкой аппаратной виртуализации.
- Оперативная память: не менее 8 Гб.
- Монитор: не менее 24" (дюймов) по диагонали.

- Наличие локальной сети со скоростью обмена не менее 1 Гб/сек.
- Наличие доступа в сеть Интернет со скоростью не менее 1 Мбит/сек.
- Наличие клавиатуры и манипулятора «мышь».

На компьютерах должно быть установлено следующее программное обеспечение:

- Операционная система: любая современная операционная система не ранее 2022 года выпуска допускающая установку JDK версии не ниже 21 и среды разработки программного обеспечения, поддерживающей язык программирования Kotlin.
- Среда разработки: IntelliJ IDEA Community Edition или Visual Studio Code.
- Среда исполнения: JDK версии не ниже 21.

4 Порядок выполнения и сдачи работы

Для выполнения лабораторной работы рекомендуется придерживаться следующего порядка выполнения:

1. Ознакомиться с темой и целями лабораторной работы
2. Изучить теоретический материал
3. Подготовить рабочее окружение
4. Разработать программный код лабораторной работы в соответствии с заданием
5. Произвести проверку работоспособности, выполнить тестирование и отладку кода
6. Проанализировать полученные результаты и примененные в ходе решения подходы
7. Выполнить самооценку и рефлекссию
8. Сдать лабораторную работу преподавателю и получить от него обратную связь

При сдаче студентом лабораторной работы основным отчетом выступает исходный код самостоятельно созданной в процессе выполнения работы программы.

Исходный код должен быть отформатирован согласно принятым для используемого языка программирования стандартам. Является желательным наличие в исходном коде комментариев, описывающих основные части программы и особенности их функционирования. В то же время студент должен быть готов объяснить работу программы в целом и каждую отдельную ее часть при полном отсутствии комментариев (например, они могут быть удалены преподавателем при сдаче работы).

Обязательным условием сдачи является умение студента самостоятельно восстановить любой участок исходного кода программы (но не более 20 строк подряд) после его удаления. Удаленный участок должен быть самостоятельно восстановлен обучающимся заново в присутствии преподавателя. При восстановлении удаленного участка кода запрещается использовать операции Undo (Отменить) текстовых редакторов или переписывание кода участка из других источников.

Исходный код сдаваемой программы должен быть представлен в электронном виде.

Сдача исходного кода ранее сдававшихся программ или программ, код которых выложен в сети Интернет, не допускается.

5 Контрольные вопросы к лабораторной работе

1. Какие основные типы данных предоставляются в Kotlin для работы с числами? Приведите примеры их использования и объясните, как они отличаются друг от друга.
2. Как в Kotlin реализовать условное выражение, которое проверяет, является ли введенное число чётным или нечётным? Напишите пример кода и объясните его работу.
3. Опишите, как работает цикл for в Kotlin. Напишите программу, которая вычисляет сумму элементов массива целых чисел, используя цикл for.
4. Как в Kotlin объявить функцию с параметрами и возвращаемым значением? Приведите пример функции, которая принимает два целых числа и возвращает их произведение.
5. Как создать и инициализировать массив строк в Kotlin? Напишите код, который выводит все элементы массива на экран.
6. Как в Kotlin обрабатывать исключения? Напишите пример, который демонстрирует обработку исключения при делении на ноль.
7. Объясните концепцию nullable типов в Kotlin. Как можно предотвратить возникновение NullPointerException при работе с такими типами?
8. Как в Kotlin выполняется интерполяция строк? Приведите пример кода, который использует переменные внутри строки.
9. Как объявить функцию с параметрами по умолчанию в Kotlin? Напишите пример функции, которая выводит приветственное сообщение, где имя пользователя является необязательным параметром.

6 Перечень использованных информационных ресурсов

1. Kotlin Docs | Kotlin Documentation [Электронный ресурс]. – URL: <https://kotlinlang.org/docs/home.html>
2. IntelliJ IDEA – the Leading Java and Kotlin IDE [Электронный ресурс]. – URL: <https://www.jetbrains.com/idea/>
3. Visual Studio Code - Code Editing. Redefined [Электронный ресурс]. – URL: <https://code.visualstudio.com/>

Редактор А.А. Литвинова

ЛР № 04779 от 18.05.01.	В набор	В печать
Объем 0,5 усл.п.л., уч.-изд.л.	Офсет.	Формат 60x84/16.
Бумага тип №3.	Заказ №	Тираж 75. Цена

Издательский центр ДГТУ

Адрес университета и полиграфического предприятия:

344010, г. Ростов-на-Дону, пл. Гагарина, 1.