

ДИСЦИПЛИНА Разработка приложений на языке Котлин
(полное наименование дисциплины без сокращений)

ИНСТИТУТ информационных технологий

КАФЕДРА информационных технологий в атомной энергетике
(полное наименование кафедры)

ВИД УЧЕБНОГО
МАТЕРИАЛА Практическая работа
(в соответствии с пп 1-11)

ПРЕПОДАВАТЕЛЬ Золотухин Святослав Александрович
(фамилия, имя, отчество)

СЕМЕСТР 5 семестр 2025 – 2026 учебный год
(указать семестр обучения, учебный год)

ЗАДАНИЕ ДЛЯ 6 ПРАКТИЧЕСКОЙ РАБОТЫ ПО ДИСЦИПЛИНЕ «РАЗРАБОТКА ПРИЛОЖЕНИЙ НА ЯЗЫКЕ КОТЛИН»

Задание 1:

Необходимо реализовать класс `TimeMeasure` с функционалом, описанным ниже (необходимые поля продумать самостоятельно). Обязательно должны быть реализованы валидации на входные параметры.

Конструкторы:

- Возможность создать `TimeMeasure`, задав часы, минуты и секунды.
- Возможность создать `TimeMeasure`, задав часы и минуты. Секунды тогда должны проставиться нулевыми.
- Возможность создать `TimeMeasure`, задав часы. Минуты и секунды тогда должны проставиться нулевыми.

Публичные методы:

- Вывести на экран установленное в классе время в формате `hh:mm:ss`
- Вывести на экран установленное в классе время в 12-часовом формате (используя `hh:mm:ss am/pm`)
- Метод, который прибавляет переданное время к установленному в `TimeMeasure` (на вход передаются только часы, минуты и секунды).

Задание 2:

Необходимо реализовать класс `UniqueString`, который хранит внутри себя строку как массив `char` и предоставляет следующий функционал:

Конструкторы:

- Создание `UniqueString`, принимая на вход массив `char`
- Создание `UniqueString`, принимая на вход `String`

Публичные методы (названия методов, входные и выходные параметры продумать самостоятельно):

- Вернуть i-ый символ строки
- Вернуть длину строки
- Вывести строку на экран
- Проверить, есть ли переданная подстрока в UniqueString (на вход подается массив char). Вернуть true, если найдена и false иначе
- Проверить, есть ли переданная подстрока в UniqueString (на вход подается String). Вернуть true, если найдена и false иначе
- Удалить из строки UniqueString ведущие пробельные символы, если они есть
- Развернуть строку (первый символ должен стать последним, а последний первым и т.д.)

Дополнительное задание:

Реализовать класс «банкомат» Atm.

Создайте класс BankAccount в пакете banking

Реализуйте основной конструктор с параметрами: accountNumber, initialBalance, ownerName

Добавьте вторичный конструктор, принимающий только accountNumber и ownerName (баланс по умолчанию = 0)

Свойства и модификаторы:

- accountNumber - только для чтения (val), public
- balance - private, с кастомным геттером, возвращающим округленное значение
- ownerName - private с кастомным сеттером, проверяющим длину имени (не менее 2 символов)

- transactionCount - private, счетчик операций

Методы:

Пополнение счета (Double),

Снятие средств (проверять достаточность баланса),

Возвращение форматированной строки с информацией о счете.

Особенности:

Прямой доступ к балансу извне запрещен,

Все изменения баланса только через методы,

Необходимо добавить private метод logTransaction() для увеличения счетчика операций

Также, добавить проверку начального баланса (не может быть отрицательным) через блок инициализации.

Требования к отчету:

Титульный лист, оглавление, текст задачи, выполненные задания (краткое описание кода реализации каждого задания с указанием листинга кода и скриншотов работы каждой программы), вывод (что было сделано в ходе выполнения работы), список использованных источников.

Оформление работ обязательно должно отвечать требованиям СМКО МИРЭА.

При защите работы необходимо ответить на несколько контрольных вопросов.

Литература:

- 1) Лекционный материал
- 2) <https://kotlinlang.org/docs/home.html>