

## ДЗ 3 Часть 1

1. Необходимо реализовать класс `Cat`.

У класса должны быть реализованы следующие приватные методы:

- `sleep()` — выводит на экран "Sleep"
- `meow()` — выводит на экран "Meow"
- `eat()` — выводит на экран "Eat"

И публичный метод:

`status()` — вызывает один из приватных методов случайным образом.

2. Необходимо реализовать класс `Student`.

У класса должны быть следующие приватные поля:

- `String name` — имя студента
- `String surname` — фамилия студента
- `int[] grades` — последние 10 оценок студента. Их может быть меньше, но не может быть больше 10.

И следующие публичные методы:

- геттер/сеттер для `name`
- геттер/сеттер для `surname`
- геттер/сеттер для `grades`
- метод, добавляющий новую оценку в `grades`. Самая первая оценка должна быть удалена, новая должна сохраниться в конце массива (т.е. массив должен сдвинуться на 1 влево).
- метод, возвращающий средний балл студента (рассчитывается как среднее арифметическое от всех оценок в массиве `grades`)

3. Необходимо реализовать класс `StudentService`.

У класса должны быть реализованы следующие публичные методы:

- `bestStudent()` — принимает массив студентов (класс `Student` из предыдущего задания), возвращает лучшего студента (т.е. который имеет самый высокий средний балл). Если таких несколько — вывести любого.
- `sortBySurname()` — принимает массив студентов (класс `Student` из предыдущего задания) и сортирует его по фамилии.

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

Конструкторы:

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

- Возможность создать TimeUnit, 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 TimeUnit (на вход передаются только часы, минуты и секунды).

5. Необходимо реализовать класс DayOfWeek для хранения порядкового номера дня недели (byte) и названия дня недели (String).

Затем в отдельном классе в методе main создать массив объектов DayOfWeek длины 7. Заполнить его соответствующими значениями (от 1 Monday до 7 Sunday) и вывести значения массива объектов DayOfWeek на экран.

Пример вывода:

1 Monday

2 Tuesday

...

7 Sunday

6. Необходимо реализовать класс AmazingString, который хранит внутри себя строку как массив char и предоставляет следующий функционал:

Конструкторы:

- Создание AmazingString, принимая на вход массив char
- Создание AmazingString, принимая на вход String

Публичные методы (названия методов, входные и выходные параметры продумать самостоятельно). Все методы ниже нужно реализовать “руками”, т.е. не прибегая к переводу массива char в String и без использования стандартных методов класса String.

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

7. Реализовать класс TriangleChecker, статический метод которого принимает три длины сторон треугольника и возвращает true, если возможно составить из них треугольник, иначе false. Входные длины сторон треугольника — числа типа

double. Придумать и написать в методе main несколько тестов для проверки работоспособности класса (минимум один тест на результат true и один на результат false)

8. Реализовать класс “банкомат” Atm.

Класс должен:

- Содержать конструктор, позволяющий задать курс валют перевода долларов в рубли и курс валют перевода рублей в доллары (можно выбрать и задать любые положительные значения)
- Содержать два публичных метода, которые позволяют переводить переданную сумму рублей в доллары и долларов в рубли
- Хранить приватную переменную счетчик — количество созданных экземпляров класса Atm и публичный метод, возвращающий этот счетчик (подсказка: реализуется через static)