Домашнее задание:

Создайте класс Person (без метода main)

Напишите в нем поля (атрибуты), описывающие характеристики человека: пол, имя, рост, вес, возраст

Проинициализируйте их какими-то значениями

Создайте в классе методы, описывающие поведение (что он может делать) человека: идти, сидеть, бежать, говорить, учить Java

В каждом методе используйте sout для вывода соответствующего названию метода сообщения

Скомпилируйте класс, чтобы проверить его на ошибки (не пытайтесь его запустить, т. к. в нем нет метода main)

Домашнее задание:

Создайте класс Wolf

объявите в нем поля: пол, кличка, вес, возраст, окрас (не присваивайте им значения по умолчанию)

реализуйте методы: идти, сидеть, бежать, выть, охотиться

в каждом методе выводите сообщение вида: "Воет", "Сидит" и т. д.

Создайте класс WolfTest с методом main

в нем создайте объект типа Wolf

в этом же классе присвойте полям Wolf какие-то значения

получите эти значения из полей и отобразите в консоли

вызовите методы объекта

Домашнее задание:

Модифицируйте класс Wolf

у всех полей класса пропишите модификатор private

а у его методов — public

для доступа к полям создайте геттеры и сеттеры

в сеттере поля age реализуйте проверку: если возраст волка > 8 лет, то выведите сообщение "Некорректный возраст". В противном случае — сохраните значение в поле

Модифицируйте класс WolfTest

с помощью сеттеров присвойте полям экземпляра класса Wolf какие-то значения

считайте эти значения из полей с помощью геттеров и отобразите в консоли

**Итоговые задачи**

* Сдавайте задания на проверку по одному
* Сопровождайте выводимые значения переменных, ввод с клавиатуры или ошибки подходящим текстом, информируя пользователя о возникающих событиях
* Используйте для форматирования кода [правила](https://topjava.ru/blog/pravila-formatirovaniya-koda-v-java2) до главы *2.6.* включительно
* Скомпилируйте и запустите свой код перед отправкой на проверку
* Перед сдачей ДЗ сверьте свое решение со списком [рекомендаций](https://docs.google.com/document/d/1VJxMqjhRilLIqguoGvb-4T9v1jGxWQwa_KgMWEG-fmE/edit#bookmark=id.6l5teo7vhjxh)
* Присылайте на проверку только тот код, в котором нет замечаний [линтера](https://topjava.ru/blog/nastroyka-sublime-text-dlya-java#9)

1. **МОДИФИЦИРУЙТЕ “КАЛЬКУЛЯТОР”**

* создайте класс *CalculatorTest* с *main(),* разместив в немкод, отвечающий за:
  + создание объектов
  + их инициализацию
  + ввод математического выражения
  + запуск вычислений
  + ввод и обработку ответа пользователя о продолжении/завершении

* создайте класс *Calculator*, отвечающий за проверку знака и вычисления
* программа должна уметь работать с любыми целыми числами
* используйте [клавиатуру](https://youtu.be/Y2iB_DwdyfM) для ввода математического выражения
* реализуйте следующий формат ввода выражения (пример):

|  |
| --- |
| Введите первое число: 2 Введите знак операции (+, -, \*, /, ^, %): ^ Введите второе число: 10 |

* выводите сообщение об ошибке при вводе неподдерживаемого знака:

|  |
| --- |
| Ошибка: операция '&' не поддерживается |

* выводите сообщение об ошибке при делении на ноль:

|  |
| --- |
| Ошибка: деление на ноль запрещено |

* после каждого вычисления выводите сообщение:

|  |
| --- |
| Хотите продолжить вычисления? [yes/no]: |

* *yes* — пользователь заново вводит математическое выражение
* *no* — программа завершается
* выводите повторно сообщение при вводе ни *yes* и ни *no*

* замените все *if-else*, проверяющие знак, на оператор [switch](https://youtu.be/QJHcGWbzk3U)
* реализуйте возведение в отрицательную степень

1. **МОДИФИЦИРУЙТЕ “JAEGER“**
   * сделайте класс *Jaeger* универсальным (без инициализации полей в самом классе, как показано в уроке)
   * информацию о роботах берите [здесь](https://pacificrim.fandom.com/wiki/Category:Jaegers)
   * выберите два разных робота, затем найдите между ними что-то общее для реализации полей и методов (делайте все максимально просто).
   * объявите два экземпляра класса *Jaeger* в классе *JaegerTest* и поэкспериментируйте с ними (проинициализируйте поля, выведите их значения, вызовите методы)
   * для инициализации полей объектов используйте для первого робота сеттеры, а для второго — [конструктор](https://topjava.ru/blog/pravila-formatirovaniya-koda-v-java2#:~:text=%D0%9A%D0%BE%D0%BD%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%82%D0%BE%D1%80%D1%8B%20%D1%81%D0%BB%D0%B5%D0%B4%D1%83%D0%B5%D1%82%20%D1%80%D0%B0%D0%B7%D0%BC%D0%B5%D1%89%D0%B0%D1%82%D1%8C%20%D0%B2%C2%A0%D0%BF%D0%BE%D1%80%D1%8F%D0%B4%D0%BA%D0%B5%2C%20%D0%B7%D0%B0%D0%B2%D0%B8%D1%81%D1%8F%D1%89%D0%B5%D0%BC%20%D0%BE%D1%82%C2%A0%D0%BF%D1%80%D0%B8%D0%BD%D0%B8%D0%BC%D0%B0%D0%B5%D0%BC%D1%8B%D1%85%20%D0%B8%D0%BC%D0%B8%20%D1%87%D0%B8%D1%81%D0%BB%D0%B0%20%D0%B0%D1%80%D0%B3%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%BE%D0%B2%3A%20%D0%BE%D1%82%C2%A0%D0%BC%D0%B5%D0%BD%D1%8C%D1%88%D0%B5%D0%B3%D0%BE%20%D0%BA%C2%A0%D0%B1%D0%BE%D0%BB%D1%8C%D1%88%D0%B5%D0%BC%D1%83) ([1](https://youtu.be/f88zS-etDWs), [2](https://topjava.ru/blog/rukovodstvo-po-konstruktoram-v-java))

1. **МОДИФИЦИРУЙТЕ “УГАДАЙ ЧИСЛО”**
   * правила игры:
     + два игрока по очереди угадывают число, загаданное компьютером, выполняя ввод с клавиатуры
     + после каждой неудачной попытки ход переходит к следующему игроку
     + победить может только один игрок
     + игра продолжается до тех пор, пока число не будет угадано
     + в игре разрешен ввод чисел в отрезке [1, 100]

* создайте класс *GuessNumberTest* с *main()* и кодом, отвечающим за:
  + создание объектов
  + их инициализацию
  + запуск игры
  + ввод и обработку ответа игрока о продолжении/завершении
* также создайте классы:
  + *Player* — описывает игроков (один экземпляр класса — один игрок)
  + *GuessNumber* — отвечает за процесс угадывания

* вводите имена игроков с клавиатуры
* используя переданные в конструктор аргументы, проинициализируйте:
  + поле *name* в классе *Player*
  + два поля типа *Player* в классе GuessNumber
* для загадывания компьютером числа используйте класс [Math.random()](https://vertex-academy.com/tutorials/ru/generaciya-sluchajnyx-chisel-v-java/)
* выводите имя победителя
* по окончанию игры выводите сообщение:

|  |
| --- |
| Хотите продолжить игру? [yes/no]: |

* *yes* — игра запускается повторно с теми же игроками и новым загаданным числом
* *no* — игра завершается
* выводите повторно сообщение при вводе ни *yes* и ни *no*

**Рекомендации**

1. Внимательно сверьте свое решение с каждым пунктом ДЗ. Если вам кажется, что в условии есть ошибка или вам что-то непонятно, то задайте вопрос наставнику

1. Внимательно читайте назначение классов. Каждый класс должен содержать только необходимый для него код

1. Класс *Player* должен состоять из:
   * конструктора, принимающего имя игрока
   * *String name* — имя игрока (одного!)
   * *int number* — введенное пользователем число
   * геттеров/сеттеров (если геттеры не используются, их можно не писать)

1. Как, используя класс *Scanner*, получить тип *char*:

|  |
| --- |
| **char** sign = scan.nextLine().charAt(0); |

1. Если *Scanner* пропускает ввод с клавиатуры (или вывод дублируется) и не дает его сделать, то ознакомьтесь с [этой информацией](https://ru.stackoverflow.com/questions/526818/scanner-%D0%BD%D0%B5-%D1%81%D1%87%D0%B8%D1%82%D1%8B%D0%B2%D0%B0%D0%B5%D1%82-%D1%81%D1%82%D1%80%D0%BE%D0%BA%D1%83-%D0%BF%D0%BE%D1%81%D0%BB%D0%B5-nextint)

**Задание на**

**именование методов**

* Создайте *четыре* класса, придумав им имена, которые содержат:
  1. не *boolean-методы*
  2. *boolean-методы*
  3. *main()* с вызовами всех методов экземпляров предыдущих классов*.* При этом поместите вызываемые методы каждого экземпляра в отдельные методы
  4. [статический](https://topjava.ru/blog/rukovodstvo-po-modifikatoru-static-v-java) метод, возвращающий имя метода, в котором он был вызван

* Вычленяйте для именования методов из описания только значимые слова, т. к. оно специально дается многословным
* Методы должны содержать только код, выводящий на консоль их названия, описание и возврат значения, если это требуется по заданию
* Методы не должны принимать аргументы и выполнять вычисления
* Отделяйте блоки в коде и в консоли пустой строкой, чтобы они не сливались
* Не используйте ввод с клавиатуры
* Одно описание соответствует одному методу
* Выводите возвращаемые *boolean-методами* значения, как показано в образце ниже

* Используйте для форматирования кода [правила](https://topjava.ru/blog/pravila-formatirovaniya-koda-v-java2#1) до главы *2.5.* включительно
* Присылайте на проверку только тот код, в котором нет замечаний [линтера](https://topjava.ru/blog/nastroyka-sublime-text-dlya-java#9)

Реализуйте методы из списка ниже в формате:

|  |
| --- |
| **public void startGameplay() {     System.out.println(получитьИмяТекущегоМетода() +**  **"() -> старт игрового процесса"); }  public boolean isExistEmail() {     System.out.print(получитьИмяТекущегоМетода() +**  **"() -> email существует? ");     return false; }** |

Вывод:

|  |
| --- |
| startGameplay() -> старт игрового процесса isExistEmail() -> email существует? false |

*Погуглите, как в Java можно получать имя текущего метода (в код можно не углубляться — лишь бы работал)*

1. **HЕ BOOLEAN-МЕТОДЫ (это название блока)**

* найти самое длинное слово в предложении из книги по *Java*
* выбрать пункт меню в текстовом редакторе на *macOS*
* вычислить среднее значение оценок в школе *№1234*
* подсчитать количество уникальных слов в *"Война и Мир"*
* вывести сообщение об ошибке
* синхронизировать данные с облачным хранилищем
* восстановить данные из резервной копии от *11.03.2024*
* приостановить загрузку mp3-файла группы "Ария"
* сбросить настройки до заводских для пылесоса *Mi*
* записать содержимое в файл по указанному пути на флешку
* преобразовать температуру из Цельсия в Фаренгейт
* ввести математическое выражение с тремя аргументами
* выявить победителя среди гонщиков игры *"Need For Speed"*
* найти книгу по имени писателя

1. **BOOLEAN-МЕТОДЫ (должны возвращать *true* или *false*)**

* программа выполняется далее или завершается?
* последовательность содержит уникальную цифру?
* пользователь ввел букву или что-то другое?
* в проверяемых числах, есть равные цифры?
* в игре "Марио" остались попытки?
* пользователь ввёл пустую строку или из одних пробелов?
* на кубике, который бросил компьютер, выпало четное число?
* путь до файла, который вы ищите на ssd, действительный?
* файл по указанному адресу существует?