1 Проект Ипподром:

1.1. Сегодня мы напишем небольшую игру под названием "***Ипподром***".  
Когда я говорю мы - я имею ввиду тебя. Я же буду работать наставником.

Для начала нам понадобятся классы "***ипподром***" и "***лошадь***".  
Создай классы **Hippodrome** (***ипподром***), **Horse** (**лошадь**).

Также не забудь, что любая программа начинается с метода main.  
Добавь его в класс **Hippodrome**.

**Требования:**

* •

Класс Hippodrome должен быть создан в отдельном файле.

* •

Класс Horse должен быть создан в отдельном файле.

* •

Класс Hippodrome должен быть объявлен с модификатором доступа public.

* •

Класс Horse должен быть объявлен с модификатором доступа public.

* •

Метод public static void main(String[] args) должен быть создан в классе Hippodrome.

1.2. Раз это ипподром, то на нем должны быть лошади.  
Значит наш ипподром должен хранить список всех его лошадей.  
Добавь поле horses типа **List**<Horse> в класс **Hippodrome**.  
А чтобы лошадей не украли - сделай это поле **приватным**.  
Добавь **getter** для этого поля.

**Требования:**

* •

В классе Hippodrome должен быть создан список лошадей(horses).

* •

В классе Hippodrome должен быть создан метод getHorses возвращающий список horses.

* •

Метод getHorses должен быть объявлен с модификатором доступа public.

* •

Поле horses должно быть объявлено с модификатором доступа private.

1.3. Несмотря на то что мы объявили переменную horses, сам список еще не создан (если ты конечно не успел опередить нас).  
Создай конструктор класса **Hippodrome** с одним параметром типа **List**.  
Сохрани в поле horses полученный в качестве параметра **список** (*инициализируй поле****horses***).

**Требования:**

* •

В классе Hippodrome должен быть создан конструктор с одним параметром типа List.

* •

Должна существовать возможность создавать объекты типа Hippodrome извне класса с помощью оператора new.

* •

Поле horses должно быть корректно инициализировано в конструкторе класса Hippodrome.

1.4. Теперь вернемся к лошадям. У каждой лошади на скачках должны быть известны имя (**name**) и скорость (**speed**).  
Наши лошади будут бежать просто определенное время (**100** ***секунд/"шагов"***).  
Будем определять победителя, как лошадь, пробежавшую наибольшую дистанцию.  
Поэтому нам понадобится хранить еще и расстояние (**distance**), которое лошадь уже пробежала.  
Добавь в класс **Horse** переменные name (String), speed (double), distance (double).

**Требования:**

* •

В классе Horse должно быть создано поле name.

* •

В классе Horse должно быть создано поле speed.

* •

В классе Horse должно быть создано поле distance.

* •

Поле name должно быть типа String.

* •

Поле speed должно быть типа double.

* •

Поле distance должно быть типа double.

1.5. Закончим написание класса лошадь (**Horse**).  
Добавь конструктор с параметрами (name, speed, distance).  
Добавь **getter'ы** и **setter'ы** для всех полей класса **Horse**.  
Делай все методы public, если явно не указано обратное.

**Требования:**

* •

В классе Horse должен быть создан конструктор с параметрами String, double и double (name, speed и distance).

* •

Конструктор класса Horse должен корректно инициализировать поля класса в соответствии с полученными параметрами.

* •

В классе Horse должны быть созданы getter'ы для всех полей класса.

* •

В классе Horse должны быть созданы setter'ы для всех полей класса.

* •

Getter'ы класса Horse должны возвращать значения соответствующих полей класса.

* •

Setter'ы класса Horse должны устанавливать значения соответствующих полей класса.

1.6. Теперь перейдем к классу **Hippodrome** и методу main.  
Нам нужно создать объект типа Hippodrome и добавить ему несколько лошадей.

Для начала, в классе **Hippodrome**:  
Создай статическое поле game типа **Hippodrome**.

В методе main требуется:  
а) Создать объект типа **Hippodrome** и сохранить его в поле game.  
б) Создать три объекта "***лошадь***". Имена придумай сам. Начальные скорость у всех лошадей - **3**, дистанция - **0**.  
в) Добавить созданных лошадей в список лошадей ипподрома (**horses**). Получить список лошадей ипподрома можно с помощью метода getHorses.

**Требования:**

* •

В классе Hippodrome должно быть создано поле game типа Hippodrome.

* •

Поле game должно быть статическим.

* •

Поле game НЕ должно быть приватным.

* •

В методе main должно инициализироваться поле game.

* •

В методе main в список лошадей ипподрома должны быть добавлены три лошади.

* •

Скорости всех лошадей должны быть равны 3, а дистанции - 0.

1.7. Но и это еще не все - надо чтобы лошади бежали.  
Добавь в класс **Hippodrome** методы run, move и print. Без параметров.  
Метод move будет управлять движением всех лошадей.  
Метод print отрисовывать их на экран.  
А метод run - управлять всем этим.

**Требования:**

* •

В классе Hippodrome должен быть создан метод run без параметров.

* •

В классе Hippodrome должен быть создан метод move без параметров.

* •

В классе Hippodrome должен быть создан метод print без параметров.

1.8. В методе run сделай цикл от **1** до **100**. Это и будет наш забег.  
В теле цикла вызываем сначала move, затем print.  
Чтобы весь цикл не отработал за долю секунды - добавь в него еще Thread.sleep(**200**);

**Требования:**

* •

В методе run класса Hippodrome должен быть 100 раз вызван метод move.

* •

В методе run класса Hippodrome должен быть 100 раз вызван метод print.

* •

В методе run класса Hippodrome должен быть 100 раз вызван метод Thread.sleep(200).

* •

Метод print должен быть вызван после метода move.

1.9. Теперь вернемся к методам move и print. Начнем с move.

В методе move класса **Hippodrome** в цикле у каждой лошади мы вызываем метод move.

Да ты прав, его еще нет у класса **Horse**.

Поэтому в класс **Horse** надо добавить свой метод move :)

И метод print, кстати тоже.

Если я не говорю ничего насчет параметров метода, значит метод без параметров.

Делай все методы **public**, если явно не указано обратное.

**Требования:**

* •

В классе Horse должен быть создан метод move.

* •

В классе Horse должен быть создан метод print.

* •

В методе move класса Hippodrome должен быть вызван метод move по одному разу для каждой лошади(каждого элемента списка horses).

1.10. Еще нужно написать метод print класса **Hippodrome**.  
В нем тоже все просто: в цикле для каждой лошади вызываем ее метод **print**.  
Ну, и еще выведи после цикла **10** пустых строк: System.out.println() - чтобы было красивее.

**Требования:**

* •

В методе print класса Hippodrome должен быть вызван метод print по одному разу для каждой лошади(каждого элемента списка horses).

* •

В методе print должны быть выведены на экран 10 пустых строк.

1.11. С классом **Hippodrome** почти закончили.  
Добавь в конец метода main вызов run().

**Подсказка:**  
run() *- это нестатический метод, поэтому вызвать его можно только у объекта.*  
*А где взять объект?*

**Подсказка 2:**  
game.run();

**Требования:**

* •

В методе main должен быть вызван метод run у объекта типа Hippodrome.

1.12 Осталось совсем немного - дописать класс **Horse**.  
Каждый ход у лошади будет вызываться метод move.  
Когда у лошади вызывают метод move, лошадь должна пробежать некоторую дистанцию.  
Дистанция зависит от скорости лошади (speed). В самом простом варианте, выглядеть этот метод должен примерно так: distance += speed;  
Но, чтобы было интереснее, давай сделаем так, чтобы скорость все время немного менялась.  
Для этого умножь speed на случайное число.  
Случайное число можно получить с помощью метода Math.random().

**Требования:**

* •

В методе move класса Horse значение поля distance должно увеличиваться на расстояние пройденное за один "ход"(значение поля speed) умноженное на случайное число от нуля до единицы полученное с помощью вызова метод Math.random().

1.13. Теперь вернемся к методу print класса **Horse**.

Т.к. мы работаем с консолью, то все лошади на бегах будут выглядеть примерно так:  
........Slevin  
....Lucky  
..........Homer

Другими словами, в методе print надо вывести на экран строку состоящую из точек и имени лошади.  
Количество точек равно distance, округленному (*в меньшую сторону*) до **целого числа**.

**Требования:**

* •

Метод print должен выводить на экран строку состоящую из точек и имени лошади. Количество точек равно целой части distance.

1.14. Запускаем и любуемся.  
У нас каждые полсекунды отображается **новый кадр** с ситуацией на ипподроме.  
Мышкой уменьши размер консоли так, чтобы был виден только один "***кадр***" и на том же самом месте.

**Тогда можно наблюдать забег в живую и даже покомментировать:**  
*- Старт.*  
*- Гомер неожиданно обходит Лаки.*  
*- Слевин вырывается вперед.*  
*- Вперед Лаки!*  
*- 10 баксов на Слевина.*  
*- Похоже Лаки сбросил жокея и расслабился.*  
*- Гомер уверенно вырывается вперед.*  
*- Неожиданно для всех побеждает Гомер. Вот это номер!*

**Требования:**

* •

Расслабься и получай удовольствие.

1.15. Добавим определение победителя.  
В классе **Hippodrome** сделаем два метода:  
public Horse getWinner() и public void printWinner()

Метод getWinner должен возвращать лошадь пробежавшую самую большую дистанцию.  
Метод printWinner выводит на экран имя победителя в виде: ***Winner is*<name>*!***

**Пример:**  
Winner is Lucky!

**Требования:**

* •

В классе Hippodrome должен быть создан метод getWinner без параметров.

* •

В классе Hippodrome должен быть создан метод printWinner без параметров.

* •

Метод getWinner должен возвращать лошадь пробежавшую наибольшую дистанцию.

* •

Метод printWinner должен выводить на экран имя победителя на экран в формате заданном в условии задачи.

1.16. Теперь уже точно все.

Добавь вызов метода printWinner в конец метода main.

Запускай и любуйся своей первой компьютерной игрой :)

**Требования:**

* •

Метод printWinner должен быть вызван в методе main после метода run.