**Некоторые воображают, что как только им удастся попасть на Луну, они сейчас же примутся прыгать по ее поверхности словно кузнечики**

+ **Некоторые объясняют это тем, что на Луне сила тяжести чуть ли не в шесть раз меньше, чем на Земле.**

**+ Незнайка и Пончик избежали этого.**

+ **Хотя Луна и притягивала их с меньшей силой, чем притягивала Земля**

**+ Они не почувствовали перемен в вес.**

**+ Это объяснялось тем, что они долгое время провели в состоянии невесомости**

**+ Они успели отвыкнуть от тяжести.**

**+ Тот вес, который они приобрели на1 Луне, показался им самым нормальным, самым обыкновенным весом, который они имели и на Земле.**

**+ Интерфейс - Во всяком случае, они не прыгали по Луне словно какие-нибудь там кузнечики или блохи, а ходили нормально.**

**+ Интерфейс - Правда, у Пончика по временам появлялось ощущение, будто все вокруг перевернуто вверх ногами. И Луна, и горы, и он сам, и Незнайка (список)**

**+ Незнайка шагал впереди**

**+ Всё казалось для Незнайка вверх тормашками.**

**+ Интерфейс Незнайка мерещилось, будто лунная поверхность вверху, а небо со всеми звездами и Солнцем внизу, и сам он висит вниз головой, прицепившись к лунной поверхности подошвами космических сапог, которые были у него на ногах.**

**+ В такие моменты он опасался, что вот-вот выскользнет из своих сапог и полетит в мировое пространство вниз головой, а сапоги останутся на Луне.**

**+ Интерфейс Это заставляло Незнайка поминутно хвататься руками за голенища космнические сапоги и потуже натягивать их на ноги.**

@startuml  
abstract class Creature {  
 + name: String  
 + position: String  
 + orientation: String  
 + getName(): String  
 + getPosition(): String  
 + getGravity(): String  
 + setName(name: String): void  
 + setName(position: String): void  
 + setName(orientation: String): void  
  
}  
  
class Neznaika {  
 + avoid(): void  
 + dontfeel(): void  
 + weightlessness(): void  
 + forget(): String  
 + walkedAhead(WalkTypes parametres): void  
 + upsideDown(UpsidePosition: parametres):void  
 + toString(): String  
 + equals(Object): boolean  
 + hashCode(): int  
}  
class Ponchik {  
 + avoid(): void  
 + dontfeel(): void  
 + weightlessness(): void  
 + forget(): String  
 + Feeling(): void  
 + toString(): String  
 + equals(Object): boolean  
 + hashCode(): int  
}  
enum WalkTypes {  
 AHEAD  
 BEHIND  
 - type: string  
 + Type(): String  
}  
enum WeightType {  
 NORMAL  
 GOOD  
 - type: string  
 + Type(): String  
}  
  
enum AttractionStrength {  
 STRONG  
 WEAK  
 EQUAL  
 - type: string  
 + Type(): String  
}  
enum UpsidePosition {  
 UP  
 DOWN  
 - type: string  
 + Type(): String  
}  
enum BootsTight {  
 TIGHT  
 - type: string  
 + Type(): String  
}  
  
enum WeightlessnessTime {  
 LONG  
 SHORT  
 - type: string  
 + Type(): String  
}  
enum GravityStrength {  
 MORE  
 LESS  
 - type: string  
 + Type(): String  
}  
enum LegsPosition {  
 UP  
 DOWN  
 - type: string  
 + Type(): String  
}  
enum Fly {  
 UP  
 DOWN  
 - type: string  
 + Type(): String  
}  
  
  
interface GetWeight {  
 + getweight(name1, p1, p2, name2): void  
}  
interface Jump {  
 + jump(name1, name2, name3, name4): void  
}  
interface Afraid {  
 + afraid(name1, name2, name3, name4): void  
}  
interface Imagine {  
 + imagine(): void  
}  
  
interface Сomparison {  
 + comparison(name1, name2): void  
}  
interface ExactСomparison {  
 + exactcomparison(name1, name2, p1, p2): void  
}  
interface Force {  
 + force(): void  
}  
  
  
  
  
  
  
  
  
  
  
  
class СreatureObj {  
 + name  
 getName(): String  
 + toString(): String  
 + equals(Object): boolean  
 + hashCode(): int  
}  
class SpaceBody {  
 + name  
 getName(): String  
 + toString(): String  
 + equals(Object): boolean  
 + hashCode(): int  
}  
  
  
''''''''''''''''''''''''  
class Location {  
 + surfaceType: String  
 + skyDescription: String  
 + getSkyDescription():  
 + getSkyDescription()  
 + toString(): String  
 + equals(Object): boolean  
 + hashCode(): int  
}  
  
class SpaceBoots {  
 + size  
 + material  
 + attachToSurface()  
 + toString(): String  
 + equals(Object): boolean  
 + hashCode(): int  
}  
  
  
class CelestialBody {  
 + name  
 + gravity  
 + getName(): String  
 + getGravity(): String  
 + toString(): String  
 + equals(Object): boolean  
 + hashCode(): int  
}  
  
  
  
Creature --> Neznaika  
Creature --> Ponchik  
  
  
Neznaika --> Imagine  
Neznaika ---> Afraid  
Neznaika --> GetWeight  
Neznaika --> Jump  
Ponchik --> GetWeight  
Ponchik -- Jump  
CelestialBody --> ExactСomparison  
CelestialBody --> Сomparison  
Neznaika --> Force  
Force --> BootsTight  
Neznaika ---> WeightlessnessTime  
Neznaika ---> WalkTypes  
Сomparison --> AttractionStrength  
ExactСomparison --> GravityStrength  
GetWeight ---> WeightType  
Ponchik ----> LegsPosition  
Neznaika ----> UpsidePosition  
Afraid --> Fly  
@enduml

@startuml  
abstract class Creature {  
 + name: String  
 + position: String  
 + orientation: String  
 + getName(): String  
 + getPosition(): String  
 + getGravity(): String  
 + setName(name: String): void  
 + setName(position: String): void  
 + setName(orientation: String): void  
  
}  
  
class Neznaika {  
 + avoid(): void  
 + dontfeel(): void  
 + weightlessness(): void  
 + forget(): String  
 + steps(): String  
 + justice(): String  
 + strong(): String  
 + thinner\_than(name): String  
 + walkedAhead(WalkTypes parametres): void  
 + upsideDown(UpsidePosition: parametres):void  
 + toString(): String  
 + equals(Object): boolean  
 + hashCode(): int  
}  
  
class BodyPart {  
 + toString(): String  
}  
  
class Ponchik {  
 + avoid(): void  
 + dontfeel(): void  
 + weightlessness(): void  
 + forget(): String  
 + understood(): String  
 + afraid\_about\_state(): String  
 + feeling\_about\_all(): String  
 + feeling\_over(): String  
 + opinion(): String  
 + threw\_up\_hands(): String  
 + went\_obediently(): String  
 + Feeling(): void  
 + toString(): String  
 + equals(Object): boolean  
 + hashCode(): int  
}  
enum WalkTypes {  
 AHEAD  
 BEHIND  
 - type: string  
 + Type(): String  
}  
enum ExwNorm {  
 NOTNORM  
 NORM  
 - type: string  
 + Type(): String  
}  
  
enum ETC\_enum {  
 LESSN  
 LESS  
 MORE  
 - type: string  
 + Type(): String  
}  
enum BodyType {  
 UP  
 DOWN  
 - type: string  
 + Type(): String  
}  
enum WeightType {  
 NORMAL  
 GOOD  
 - type: string  
 + Type(): String  
}  
enum Extra {  
 EXTRA  
 - type: string  
 + Type(): String  
}  
  
enum Clarification {  
 VERY  
 NOTVERY  
 - type: string  
 + Type(): String  
}  
enum AttractionStrength {  
 STRONG  
 WEAK  
 EQUAL  
 - type: string  
 + Type(): String  
}  
enum UpsidePosition {  
 UP  
 DOWN  
 - type: string  
 + Type(): String  
}  
enum BootsTight {  
 TIGHT  
 - type: string  
 + Type(): String  
}  
  
enum WeightlessnessTime {  
 LONG  
 SHORT  
 - type: string  
 + Type(): String  
}  
enum GravityStrength {  
 MORE  
 LESS  
 - type: string  
 + Type(): String  
}  
enum LegsPosition {  
 UP  
 DOWN  
 - type: string  
 + Type(): String  
}  
enum Light {  
 LG  
 DARK  
 - type: string  
 + Type(): String  
}  
enum Fly {  
 UP  
 DOWN  
 - type: string  
 + Type(): String  
}  
enum State {  
 STRONG  
 FAT  
 - type: string  
 + Type(): String  
}  
  
  
interface GetWeight {  
 + getweight(name1, p1, p2, name2): void  
}  
interface Jump {  
 + jump(name1, name2, name3, name4): void  
}  
interface Afraid {  
 + afraid(name1, name2, name3, name4): void  
}  
interface Imagine {  
 + imagine(): void  
}  
  
interface Сomparison {  
 + comparison(name1, name2): void  
}  
interface ExactСomparison {  
 + exactcomparison(name1, name2, p1, p2): void  
}  
interface Force {  
 + force(): void  
}  
  
interface Blood {  
 + blood(): void  
}  
  
interface Justice {  
 + justice(): void  
}  
  
  
class СreatureObj {  
 + name  
 getName(): String  
 + toString(): String  
 + equals(Object): boolean  
 + hashCode(): int  
}  
class SpaceBody {  
 + name  
 getName(): String  
 + toString(): String  
 + equals(Object): boolean  
 + hashCode(): int  
}  
  
  
''''''''''''''''''''''''  
class Location {  
 + name  
 + getName(): String  
 + toString(): String  
 + equals(Object): boolean  
 + hashCode(): int  
}  
  
interface MountainUP {  
 + mountain\_up(name1, name3): void  
}  
enum Distance {  
 NORFAR  
 FAR  
 - type: string  
 + Type(): String  
}  
enum Cloud {  
 CLOUD  
 - type: string  
 + Type(): String  
}  
interface Road {  
 + road(name1): void  
}  
  
class SpaceBoots {  
 + size  
 + material  
 + attachToSurface()  
 + toString(): String  
 + equals(Object): boolean  
 + hashCode(): int  
}  
  
  
class CelestialBody {  
 + name  
 + gravity  
 + getName(): String  
 + getGravity(): String  
 + toString(): String  
 + equals(Object): boolean  
 + hashCode(): int  
}  
  
  
  
Creature --> Neznaika  
Creature --> Ponchik  
  
  
Neznaika --> Imagine  
Neznaika ---> Afraid  
Neznaika --> GetWeight  
Neznaika --> Jump  
Ponchik --> GetWeight  
Ponchik -- Jump  
  
  
CelestialBody --> ExactСomparison  
CelestialBody --> Сomparison  
Neznaika --> Force  
Force --> BootsTight  
Neznaika ---> WeightlessnessTime  
Neznaika ---> WalkTypes  
Сomparison --> AttractionStrength  
ExactСomparison --> GravityStrength  
GetWeight ---> WeightType  
Ponchik ----> LegsPosition  
Neznaika ----> UpsidePosition  
Afraid --> Fly  
BodyPart --> MountainUP  
BodyPart --> Road  
MountainUP --> Distance  
MountainUP --> Cloud  
Neznaika ---> State  
Neznaika --> Clarification  
Location --> Light  
Neznaika ----> Justice  
@enduml