

Trabajo Práctico # 2

Estructuras de Datos, Universidad Nacional de Quilmes

29 de agosto de 2014

Aclaraciones:

- *Los ejercicios fueron pensados para ser resueltos en el orden en que son presentados. No se saltee ejercicios sin consultar antes a un docente.*
- *Recuerde que puede aprovechar en todo momento las funciones que ha definido, tanto las de esta misma práctica como las de prácticas anteriores.*
- *Pruebe todas sus implementaciones, al menos en una consola interactiva.*
- *Es sumamente aconsejable resolver los ejercicios utilizando primordialmente los conceptos y metodologías vistos en clase, dado que los exámenes de la materia evaluarán principalmente este aspecto. Si se encuentra utilizando formas alternativas al resolver los ejercicios consulte a los docentes.*
- *No dude en manifestar observaciones y críticas sobre los ejercicios de esta práctica, que con gusto serán recibidas por los docentes.*
- *Los ejercicios del anexo pueden obviarse, pero recuerde que aportan una comprensión más profunda sobre los temás que aborda esta práctica. Considere resolverlos si se encuentra practicando para una instancia de evaluación y ya resolvió todos los anteriores.*

1. Tipos de datos definidos por el usuario

1. Definir el tipo de dato `Dir`, con las alternativas Norte, Sur, Este y Oeste. Luego implementar las siguientes funciones:

- `opuesto :: Dir -> Dir`
Dada una dirección devuelve su opuesta
- `siguiente :: Dir -> Dir`
Dada una dirección devuelve su siguiente, en sentido horario.

2. Definir el tipo de dato `Persona`, como un nombre la edad de la persona. Realizar las siguientes funciones:

- `nombre :: Persona -> String`
Devuelve el nombre de una persona
- `edad :: Persona -> Int`
Devuelve la edad de una persona
- `crecer :: Persona -> Persona`
Dada una persona la devuelve con su edad aumentada en 1.
- `cambioDeNombre :: String -> Persona -> Persona`
Dados un nombre y una persona, reemplaza el nombre de la persona por este otro.
- `esMenorQueLaOtra :: Persona -> Persona -> Bool`
Dadas dos personas indica si la primera es más joven que la segunda.
- `mayoresA :: Int -> [Persona] -> [Persona]`
Dados una edad y una lista de personas devuelve todas las personas que son mayores a esa edad.

- `promedioEdad :: [Persona] -> Int`
Dada una lista de personas devuelve el promedio de edad entre esas personas. La lista al menos posee una persona.
 - `elMasViejo :: [Persona] -> Persona`
Dada una lista de personas devuelve la persona más vieja de la lista. La lista al menos posee una persona.
3. Definir los tipos de datos *Pokemon*, como un *TipoDePokemon* (agua, fuego o planta) y un porcentaje de energía; y *Entrenador*, como un nombre y una lista de *Pokemon*. Luego definir las siguientes funciones:
- `elementoGanador :: TipoDePokemon -> TipoDePokemon`
Dado un *TipoDePokemon* devuelve el elemento que le gana a ese. Agua le gana a fuego, fuego a planta y planta a agua.
 - `leGanaA :: Pokemon -> Pokemon -> Bool`
Dados dos pokemon indica si el primero le puede ganar al segundo. Se considera que gana si su elemento es opuesto al del otro pokemon. Si poseen el mismo elemento se considera que no gana.
 - `capturarPokemon :: Pokemon -> Entrenador -> Entrenador`
Agrega un pokemon a la lista de pokemon del entrenador.
 - `cantidadDePokemons :: Entrenador -> Int`
Devuelve la cantidad de pokemons que posee el entrenador.
 - `cantidadDePokemonsDeTipo :: TipoDePokemon -> Entrenador -> Int`
Devuelve la cantidad de pokemons de determinado tipo que posee el entrenador.
 - `lePuedeGanar :: Entrenador -> Pokemon -> Bool`
Dados un entrenador y un pokemon devuelve *True* si el entrenador posee un pokemon que le gane ese pokemon.
 - `puedenPelear :: TipoDePokemon -> Entrenador -> Entrenador -> Bool`
Dados un tipo de pokemon y dos entrenadores, devuelve *True* si ambos entrenadores tiene al menos un pokemon de ese tipo y que tenga energía para pelear.
 - `esExperto :: Entrenador -> Bool`
Dado un entrenador devuelve *True* si ese entrenador posee al menos un pokemon de cada tipo posible.
 - `fiestaPokemon :: [Entrenador] -> [Pokemon]`
Dada una lista de entrenadores devuelve una lista con todos los pokemon de cada entrenador.