## Робота №5

*Екземпляри класів типів*

Визначення класу типів Num, як воно задано в модулі Prelude.

*-- | Basic numeric class.*

***class*** Num a ***where***

{-# MINIMAL (+), (\*), abs, signum, fromInteger, (negate | (-)) #-}

(+), (-), (\*) :: a -> a -> a

*-- | Unary negation.*

negate :: a -> a

*-- | Absolute value.*

abs :: a -> a

*-- | Sign of a number.*

signum :: a -> a

*-- | Conversion from an 'Integer'.*

fromInteger :: Integer -> a

x - y = x + negate y

negate x = 0 - x

*-- The functions 'abs' and 'signum' should satisfy the law:*

*--*

*-- > abs x \* signum x == x*

*--*

Тип AbstractInteger один з варіантів представлення цілих чисел, він має три конструктора: Zero (константа) , Succ AbstractInteger (наступник абстрактного цілого) і Pred AbstractInteger(попередник абстрактного цілого).

***data*** AbstractInteger

= Zero

| Succ AbstractInteger

| Pred AbstractInteger

***deriving*** (Show, Eq)

Всі операції над AbstractInteger повинні підтримувати інваріант

*Дане типа AbstractInteger або Zero або результат використання тільки одного з конструкторів Succ / Pred, а не обох. Так Pred (Pred Zero) – коректне дане типа AbstractInteger, але Pred (Succ Zero) – ні, тому що його можна нормалізувати до Zero.*

Кватерніони – одне з узагальнень комплексних чисел, побудовані вперше математиком Гамільтоном для полегшення рішень проблем динаміки.

***data*** Quaternion = Quaternion Double Double Double Double

***deriving*** ( Eq)

Комплексні числа мають дві компоненти (дійсну і уявну). Кватерніони мають чотири компоненти: дійсну і три уявні – i, j, k. Уявні компоненти зв»язані наступними співвідношеннями:

*i \* i = -1 j \* j = -1 k \* k = -1*

*i \* j = k j \* i = -k k \* j = -i*

*k \* i = j i \* k = -j j \* k = i*

Всі кватерніони мають чотири компоненти. Компоненти кватерніонів – Double.

Абсолютна величина кватерніона – число, дійсна компонента якого є корінь квадратний з суми квадратів коефіцієнтів всіх компонент, а уявні рівні 0.

«Знак» (signum) кватерніона – це початкове число масштабоване дійсним компонентом абсолютної величини (тобто кожний компонент ділиться на одне і те саме число масштабування) .

Додавання – по компонентне, для множення використовуються наведені раніше співвідношення.

Для Quaternion немає розумної реалізації його як екземпляру класу Ord.

В допоміжному файлі, котрий включає визначення типів, необхідно надати визначення вказаних функцій і встановіть типи екземплярами необхідних класів.

1. Встановіть *AbstractInteger* екземпляром *типів Ord*, визначивши власну функцію порівняння.
2. Функція *aiToInteger ai,* котра перетворює *ai* дане типу AbstractInteger в Integer Наприклад:
   * aiToInteger Zero = 0
   * aiToInteger Pred (Pred Zero) = -2
3. Функція *plusAbs ai1 ai2,* котра додає *ai1* і *ai2 -* дані типу AbstractInteger, її результат дане типу AbstractInteger. Не визначайте її шляхом трансформування додавання для звичайних Integer. Наприклад:
   * plusAbs (Pred (Pred Zero)) (Succ (Succ Zero)) = Zero
   * plusAbs (Pred Zero) (Succ (Succ Zero)) = Succ Zero
4. Функція *timesAbs ai1 ai2t*, котра множить *ai1* і *ai2 -* дані типу AbstractInteger, її результат дане типу AbstractInteger. Наприклад:
   * timesAbs (Pred (Pred Zero)) (Pred (Pred (Pred Zero))) =

*=* Succ( Succ ( Succ (Succ (Succ (Succ Zero)))))

1. Встановіть *AbstractInteger* екземпляром класу *типів Num* шляхом визначення відповідних функцій для цього класу.
2. Функція *factorial n*, що обчислює факторіал *n*. Функція працює з довільними типами класів Eq і Num. Наприклад:
   * factorial (Succ (Succ (Succ Zero)))

= Succ( Succ ( Succ (Succ (Succ (Succ Zero)))))

* + factorial Zero = Succ Zero

1. Встановіть Quaternion екземпляром класу Show.
   * show (Quaternion 4 5 (-6.89) 7) = “4.0+5.0i-6.89j+7.0k”.
2. Функція *plusQuaternion q1 q2*, котра додає *q1* і *q2* дані типу Quaternion, її результат дане типу Quaternion. Наприклад
   * plusQuaternion (Quaternion 1 1 1 1) (Quaternion 2 2 2 6) = 3.0+3.0i+3.0j+7.0k
3. Функція *timesQuaternion q1 q2*, котра множить *q1* і *q2* дані типу Quaternion, її результат дане типу Quaternion. Наприклад
   * timesQuaternion (Quaternion 1 1 1 1) (Quaternion 2 2 2 6) = -8.0+8.0i+0.0j+8.0k
4. Встановіть *Quaternion* екземпляром класу *типів Num*.

*aiToInteger*  :: AbstractInteger -> Integer

*plusAbs*  :: AbstractInteger -> AbstractInteger -> AbstractInteger

*timesAbs*  :: AbstractInteger -> AbstractInteger -> AbstractInteger

*factorial*  :: (Eq a, Num a) => a -> a

*plusQuaternion*  :: Quaternion -> Quaternion -> Quaternion

*timesQuaternion* :: Quaternion -> Quaternion -> Quaternion

Зауваження:

Назва файлу Family05.hs (Family – прізвище студента). Файл включає модуль Family05 і створюється на основі файла-заготовки HWP05.hs