

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ  
Учреждение образования «Полоцкий государственный университет»

Факультет информационных технологий  
Кафедра технологий программирования

**Практическая работа**  
по дисциплине: **«Функциональное программирование»**  
на тему: **«Работа в интерпретатора Haskell»**

ВЫПОЛНИЛ

студент группы 16 ИТ-3  
Яблонский А.С.

ПРОВЕРИЛ

преподаватель  
Попкова Д.В.

Полоцк, 2018 г

## Структуры

База данных транспортных средств предназначена для хранения одного из следующих т/с: автомобиль, поезд, самолёт, корабль, дирижабль. Каждое из данных полей содержит следующие атрибуты:

1. Автомобиль
  - a. Масса
  - b. Стоимость
  - c. Вместимость
  - d. Расход топлива
2. Поезд
  - a. Кол-во вагонов
  - b. Тип (электро, дизель др)
  - c. Вместимость
3. Самолет
  - a. Вместимость
  - b. Макс. дистанция полёта
  - c. Крейсерская скорость
  - d. Производитель
4. Корабль
  - a. Вместимость
  - b. Водоизмещение
  - c. Максимальный ход
  - d. Тип(лайнер, паром, военный и др.)
5. Дирижабль
  - a. Объем
  - b. Грузоподъемность
  - c. Макс. дистанция полёта
  - d. Дата создания

Транспортные средства могут характеризоваться скоростью, вместимостью, массой, грузоподъемностью и др.

## Поля

Транспорт	Поле	Тип
Автомобиль	Масса	Int
	Стоимость	Double
	Вместимость	Int
	Расход топлива	Double
Поезд	Кол-во вагонов	Int

	Тип (электро, дизель др)	Char
	Вместимость	Int
Самолет	Вместимость	Int
	Макс. дистанция полёта	Int
	Крейсерская скорость	Int
	Производитель	String
Корабль	Вместимость	Int
	Водоизмещение	Int
	Максимальный ход	Int
	Тип(лайнер, паром, военный и др.)	Char
Дирижабль	Объем	Int
	Грузоподъемность	Int
	Макс. дистанция полёта	Int
	Дата создания(миллисекунды)	Int

## Функции

Для работы с данной базой транспортных средств, разработаны следующие методы.

1. **getCapacity** - В качестве параметров принимает любое транспортное средство. Возвращает значение поля *вместимость* транспортного средства. Если данное т/с не имеет вместимости - вернет -1.

```
getCapacity :: Transport -> Int
getCapacity t
  | (isCar t) || (isTrain t) || (isShip t) ||
  (isPlain t) = capacity t
  | otherwise = -1
```

Пример использования:

```
getCapacity (Plain 12 12 123 "Hello")
getCapacity (Car 12 12.3 4 12.2)
```

2. **getSpeed** - В качестве параметров принимает любое транспортное средство. Возвращает значение поля *скорость* транспортного средства. Если данное т/с не имеет параметра скорость - вернет -1.

```
getSpeed :: Transport -> Int
getSpeed t
  | isPlain t = speed t
  | otherwise = -1
```

Пример использования:

```
getSpeed (Plain 12 12 123 "Hello")
getSpeed (Car 12 12.3 4 12.2)
```

3. **getMaxCapacity** - В качестве параметров принимает список транспортных средств. Возвращает т/с из этого списка, которое обладает наибольшей скоростью. Генерирует ошибку, если передан пустой список. Если же список содержит несколько т/с с одинаковой вместимостью - будет возвращен первый элемент.

```
getMaxCapacity :: [Transport] -> Transport
getMaxCapacity tList
    | length tList == 0 = error "List cannot be empty"
    | otherwise = _getMaxCapacity tList (head tList)
_getMaxCapacity tList tMax
    | (length tList) == 0 = tMax
    | otherwise = if(getCapacity (head tList) > getCapacity
tMax)
                        then (_getMaxCapacity (tail tList) (head tList))
                        else (_getMaxCapacity (tail tList) tMax)
```

Пример использования:

```
getMaxCapacity [(Plain 12 12 123 "Hello"), (Plain 12 12 1232
"Plain 2")]
```

4. **getAirshipsByDate** - Возвращает список дирижаблей, который были созданы в указанные промежутки времени. В качестве параметров принимает список транспортных средств и две даты в миллисекундах. Возвращает список дирижаблей, дата создания которых входит в указанный диапазон. Если дирижаблей найдено не было - пустой список.

```
getAirshipsByDate :: [Transport] -> Int -> Int -> [Transport]
getAirshipsByDate tList sDate eDate = _getAirshipDate tList
sDate eDate []

_getAirshipDate :: [Transport] -> Int -> Int -> [Transport] ->
[Transport]
_getAirshipDate (t:tl) sDate eDate resL
    | isAirship t = if
((getCreateDate t) >= sDate && (getCreateDate t) <= eDate)
then continue (resL++[t]) else continue resL
    | otherwise = continue resL
where continue l = if (length tl
== 0) then l else _getAirshipDate tl sDate eDate l
```

Пример использования:

```
getAirshipsByDate [(Airship 12 12 123 12000), (Airship 12 12
1232 20000)] 10000 15000
```

Также несколько вспомогательных методов для определения типа транспортного средства: isCar, isTrain, isPlain, isShip и isAirship. Все они возвращают Boolean значение в зависимости от транспортного средства в параметрах.