

Языки программирования (осень 2018)

В начало ► Мои курсы ► ЯП 2018 ► Оценка лабораторных работ ► Лабораторная работа №4 ► Работа

Лабораторная работа №4

Моя работа

Инструкции для работы ▼

Для того, чтобы отправить работу на оценку, нажмите "Начало подготовки Вашей работы".

На открывшейся странице:

- в поле **Название** появившегося окна укажите точное название загружаемого файла;
- поле **Содержимое работы** оставьте пустым;
- из папки с решением перетащите загружаемый файл в поле **Приложение** или загрузите файл в это поле, используя кнопку "Добавить.." в меню этого поля;
- выполнив перечисленные пункты нажмите кнопку "Сохранить".

При необходимости, пока не окончена фаза представления работ, можно откорректировать представление работы нажав кнопку "Редактировать работу"

Lab4.hs

представлено: Среда, 7 Ноябрь 2018, 17:12

-  Lab4.hs



Самооценка

от Максим Кулаков

Оценка: 97,92 из 100,00

Форма оценки ▼

Критерий 1

Функция `funName` должна иметь вид

```
funName (Fun name _ _) = name
funName e = error $ "The expression " ++ show (e) ++ " is not a function"
```

- Решение заслуживает не более 1 балла, если оно сложнее, чем приведенное выше или, в частности, в нем определяется больше 2-х шаблонов.
- Не следует снижать оценку, если присутствуют имена параметров конструктора `Fun`, которые не используются.

- Ваша оценка должна быть беспристрастной. Вполне нормально, если решение отличается от приведенных выше вариантов. Вы проверяете правильность решения и его стиль, а не степень совпадения решения с приведенными вариантами.

Оценка для Критерий 1

5

Комментарий к Критерий 1

Критерий 2

Функция `funArg` должна иметь вид

```
funArg (Fun _ arg _) = arg
funArg e = error $ "The expression " ++ show (e) ++ " is not a function"
```

- Решение заслуживает не более 1 балла, если оно сложнее, чем приведенное выше или, в частности, в нем определяется больше 2-х шаблонов.
- Не следует снижать оценку, если присутствуют имена параметров конструктора `Fun`, которые не используются.
- Ваша оценка должна быть беспристрастной. Вполне нормально, если решение отличается от приведенных выше вариантов. Вы проверяете правильность решения и его стиль, а не степень совпадения решения с приведенными вариантами.

Оценка для Критерий 2

5

Комментарий к Критерий 2

Критерий 3

Функция `funBody` должна иметь вид

```
funBody (Fun _ _ body) = body
funBody e = error $ "The expression " ++ show (e) ++ " is not a function"
```

- Решение заслуживает не более 1 балла, если оно сложнее, чем приведенное выше или, в частности, в нем определяется больше 2-х шаблонов.
- Не следует снижать оценку, если присутствуют имена параметров конструктора `Fun`, которые не используются.
- Ваша оценка должна быть беспристрастной. Вполне нормально, если решение отличается от приведенных выше вариантов. Вы проверяете правильность решения и его стиль, а не степень совпадения решения с приведенными вариантами.

Оценка для Критерий 3

5

Комментарий к Критерий 3

Критерий 4

Функция `pairHead` должна иметь вид

```
pairHead (Pair e1 _) = e1
pairHead e = error $ "The expression " ++ show (e) ++ " is not a pair"
```

- Решение заслуживает не более 1 балла, если оно сложнее, чем приведенное выше или, в частности, в нем определяется больше 2-х шаблонов.
- Не следует снижать оценку, если присутствуют имена параметров конструктора `Pair`, которые не используются.
- Ваша оценка должна быть беспристрастной. Вполне нормально, если решение отличается от приведенных выше вариантов. Вы проверяете правильность решения и его стиль, а не степень совпадения решения с приведенными вариантами.

Оценка для Критерий 4

5

Комментарий к Критерий 4

Критерий 5

Функция `pairTail` должна иметь вид

```
pairTail (Pair _ e2) = e2
pairTail e = error $ "The expression " ++ show (e) ++ " is not a pair"
```

- Решение заслуживает не более 1 балла, если оно сложнее, чем приведенное выше или, в частности, в нем определяется больше 2-х шаблонов.
- Не следует снижать оценку, если присутствуют имена параметров конструктора `Pair`, которые не используются.
- Ваша оценка должна быть беспристрастной. Вполне нормально, если решение отличается от приведенных выше вариантов. Вы проверяете правильность решения и его стиль, а не степень совпадения решения с приведенными вариантами.

Оценка для Критерий 5

5

Комментарий к Критерий 5

Критерий 6

Функция `closureFun` должна иметь вид

```
closureFun (Closure _ f) = f
closureFun e = error $ "The expression " ++ show (e) ++ " is not a closure"
```

- Решение заслуживает не более 1 балла, если оно сложнее, чем приведенное выше или, в частности, в нем определяется больше 2-х шаблонов.
- Не следует снижать оценку, если присутствуют имена параметров конструктора `Closure`, которые не используются.
- Ваша оценка должна быть беспристрастной. Вполне нормально, если решение отличается от приведенных выше вариантов. Вы проверяете правильность решения и его стиль, а не степень совпадения решения с приведенными вариантами.

Оценка для Критерий 6

5

Комментарий к Критерий 6

Критерий 7

Функция `closureEnv` должна иметь вид

```
closureEnv (Closure env _) = env
closureEnv e = error $ "The expression " ++ show (e) ++ " is not a closure"
```

- Решение заслуживает не более 1 балла, если оно сложнее, чем приведенное выше или, в частности, в нем определяется больше 2-х шаблонов.
- Не следует снижать оценку, если присутствуют имена параметров конструктора `Closure`, которые не используются.
- Ваша оценка должна быть беспристрастной. Вполне нормально, если решение отличается от приведенных выше вариантов. Вы проверяете правильность решения и его стиль, а не степень совпадения решения с приведенными вариантами.

Оценка для Критерий 7

5

Комментарий к Критерий 7

Критерий 8

Функция `convertListToMUPL` может иметь вид

```
convertListToMUPL [] = Unit
convertListToMUPL (x : xs) = Pair x $ convertListToMUPL xs
```

или

```
convertListToMUPL l = foldr Pair Unit l
```

Оба вида решения являются правильными.

- Следует снизить оценку на балл, если решение сложнее тех, что приведены выше.
- Ваша оценка должна быть беспристрастной. Вполне нормально, если решение отличается от приведенных выше вариантов. Вы проверяете правильность решения и его стиль, а не степень совпадения решения с приведенными вариантами.

Оценка для Критерий 8

5

Комментарий к Критерий 8

Критерий 9

Функция `convertListFromMUPL` может иметь вид

```
convertListFromMUPL Unit = []  
convertListFromMUPL (Pair x xs) = x : convertListFromMUPL xs
```

- Следует снизить оценку на балл, если решение сложнее приведенного выше.
- Ваша оценка должна быть беспристрастной. Вполне нормально, если решение отличается от приведенных выше вариантов. Вы проверяете правильность решения и его стиль, а не степень совпадения решения с приведенными вариантами.

Оценка для Критерий 9

5

Комментарий к Критерий 9

Критерий 10

Реализацию функции `evalUnderEnv` будем оценивать по частям: будем выставять отдельно оценки за реализацию вычислений определенных выражений на MUPL.

Оценим реализацию вычисления значения выражения `Add`. Реализация `evalUnderEnv (Add e1 e2)` должна иметь вид

```
evalUnderEnv (Add e1 e2) env =  
  IntNum $ valOfIntNum (evalUnderEnv e1 env)  
    + valOfIntNum (evalUnderEnv e2 env)
```

- Могут использоваться локальные имена для значений подвыражений. Снижать за это оценку не нужно.
- В реализации должно быть ровно два рекурсивных вызова функции `evalUnderEnv`. Если это не так, то за решение следует поставить один балл.
- Следует снизить оценку на 1 балл, если в решении не используется `valOfIntNum`.
- Стоит снизить оценку минимум на один балл, если решение сложнее, чем приведенное выше.
- Ваша оценка должна быть беспристрастной. Вполне нормально, если решение отличается от приведенных выше вариантов. Вы проверяете правильность решения и его стиль, а не степень совпадения решения с приведенными вариантами.

Оценка для Критерий 10

5

Критерий 11

Оценим реализацию вычисления значения выражения `IfGreater`. Реализация

`evalUnderEnv (IfGreater e1 e2 e3 e4)` должна иметь вид

```
evalUnderEnv (IfGreater e1 e2 e3 e4) env =  
  if valOfIntNum (evalUnderEnv e1 env) > valOfIntNum (evalUnderEnv e2 env)  
  then evalUnderEnv e3 env  
  else evalUnderEnv e4 env
```

- Могут использоваться локальные имена для значений подвыражений `e1` и `e2`. Снижать за это оценку не нужно.
- Следует поставить за решение не более одного балла, если вводится локальное имя для значения выражения `e3` или `e4` до сравнения значений `e1` и `e2`.
- В реализации должно быть четыре рекурсивных вызова функции `evalUnderEnv`, причем при любом ходе вычислений должно выполниться не более трех таких вызовов. Если это не так, то за решение следует поставить один балл.
- Стоит снизить оценку минимум на один балл, если решение сложнее, чем приведенное выше.
- Следует снизить оценку на 1 балл, если в решении не используется `valOfIntNum`.
- Ваша оценка должна быть беспристрастной. Вполне нормально, если решение отличается от приведенных выше вариантов. Вы проверяете правильность решения и его стиль, а не степень совпадения решения с приведенными вариантами.

Оценка для Критерий 11

5

Комментарий к Критерий 11

Критерий 12

Оценим реализацию вычисления значения выражения `Pair`. Реализация `evalUnderEnv (Pair e1 e2)` должна иметь вид

```
evalUnderEnv (Pair e1 e2) env =  
  Pair (evalUnderEnv e1 env) (evalUnderEnv e2 env)
```

- Могут использоваться локальные имена для значений подвыражений `e1` и `e2`. Снижать за это оценку не нужно.
- В реализации должно быть ровно два рекурсивных вызова функции `evalUnderEnv`. Если это не так, то за решение следует поставить один балл.
- Стоит снизить оценку минимум на один балл, если решение сложнее, чем приведенное выше.
- Ваша оценка должна быть беспристрастной. Вполне нормально, если решение отличается от приведенных выше вариантов. Вы проверяете правильность решения и его стиль, а не степень совпадения решения с приведенными вариантами.

Оценка для Критерий 12

5

Критерий 13

Оценим реализацию вычисления значения выражений `Head` и `Tail`. Реализация `evalUnderEnv` для таких выражений должна иметь вид

```
evalUnderEnv (Head e) env = pairHead (evalUnderEnv e env)
evalUnderEnv (Tail e) env = pairTail (evalUnderEnv e env)
```

- Могут использоваться локальные имена для значений подвыражений. Снижать за это оценку не нужно.
- В правой части каждого равенства должен присутствовать точно один рекурсивный вызов функции `evalUnderEnv`. Если это не так, то за решение следует поставить один балл.
- Стоит снизить оценку минимум на один балл, если решение сложнее, чем приведенное выше.
- Следует снизить оценку на 1 балл, если в решении не используется `pairHead` или `pairTail`.
- Ваша оценка должна быть беспристрастной. Вполне нормально, если решение отличается от приведенных выше вариантов. Вы проверяете правильность решения и его стиль, а не степень совпадения решения с приведенными вариантами.

Оценка для Критерий 13

5

Комментарий к Критерий 13

Критерий 14

Оценим реализацию вычисления значения выражения `Fun`. Реализация `evalUnderEnv` для такого выражения должна иметь вид

```
evalUnderEnv (f @ (Fun _ _ _)) env = Closure env f
```

- Не должно производиться никаких вычислений тела функции или аргументов функции. Если такие вычисления присутствуют, следует поставить за решение не более одного балла.
- Не должно присутствовать никаких рекурсивных вызовов функции `evalUnderEnv`. Если это не так, то за решение следует поставить один балл.
- Стоит снизить оценку на один балл, если в правой части равенства присутствует вызов конструктора `Fun`.
- Стоит снизить оценку минимум на балл, если решение сложнее, чем приведенное выше.
- Ваша оценка должна быть беспристрастной. Вполне нормально, если решение отличается от приведенных выше вариантов. Вы проверяете правильность решения и его стиль, а не степень совпадения решения с приведенными вариантами.

Оценка для Критерий 14

4

Комментарий к Критерий 14

присутствует вызов конструктора

Критерий 15

Оценим реализацию вычисления значения выражения `IsAUnit`. Реализация `evalUnderEnv (IsAUnit e)` должна иметь вид

```
evalUnderEnv (IsAUnit e) env =  
  case evalUnderEnv e env of  
    Unit -> IntNum 1  
    _     -> IntNum 0
```

- Может использоваться локальное имя для значения подвыражения. Снизать за это оценку не нужно.
- В правой части равенства должен присутствовать точно один рекурсивный вызов функции `evalUnderEnv`. Если это не так, то за решение следует поставить один балл.
- Вместо сравнения с шаблоном может использоваться операция сравнения на равенство `==` со значением `Unit`. Снизать за это оценку не следует.
- Стоит снизить оценку минимум на балл, если решение сложнее, чем приведенное выше.
- Ваша оценка должна быть беспристрастной. Вполне нормально, если решение отличается от приведенных выше вариантов. Вы проверяете правильность решения и его стиль, а не степень совпадения решения с приведенными вариантами.

Оценка для Критерий 15

4

Комментарий к Критерий 15

решение сложнее, чем заданное в критерии + пробелы

Критерий 16

Оценим реализацию вычисления значения выражения `Let`. Реализация `evalUnderEnv` для такого выражения должна иметь вид

```
evalUnderEnv (Let (name, e1) e2) env =  
  evalUnderEnv e2 $ (name, evalUnderEnv e1 env) : env
```

- В решении должны присутствовать точно два рекурсивных вызова `evalUnderEnv`. Если это не так, за решение следует поставить 1 балл.
- Выражения `e1` и `e2` должны быть вычислены в разных окружениях. Для вычисления `e2` в окружение должно быть добавлено значение выражения `e1`. Если это не так, поставьте за решение не более одного балла.
- Может использоваться локальное имя для значения подвыражения `e1`. Снизать за это оценку не нужно.
- Стоит снизить оценку минимум на балл, если решение сложнее, чем приведенное выше.
- Ваша оценка должна быть беспристрастной. Вполне нормально, если решение отличается от приведенных выше вариантов. Вы проверяете правильность решения и его стиль, а не степень совпадения решения с приведенными вариантами.

Оценка для Критерий 16

5

Комментарий к Критерий 16

Критерий 17

Оценим реализацию вычисления значения выражения `Call`. Реализация `evalUnderEnv` для такого выражения должна иметь вид

```
evalUnderEnv (Call e1 e2) env =
  let
    c1 = evalUnderEnv e1 env
    fn = closureFun c1
    env1 = (funArg fn, evalUnderEnv e2 env) : closureEnv c1
    fname = funName fn
    newEnv = if fname == "" then env1 else (fname, c1) : env1
  in evalUnderEnv (funBody fn) newEnv
```

- В решении должны присутствовать точно три рекурсивных вызова `evalUnderEnv`. Если вызовов меньше, то за решение следует поставить 1 балл. Если больше, - следует снизить оценку за решение минимум на один балл.
- Следует поставить за решение не более 1 балла, если не предполагается, что результатом вычисления `e1` будет не замыкание.
- Следует снизить оценку на два балла, если при пустом имени функции в окружение для вычисления тела функции помещается ее замыкание.
- Может использоваться дополнительное локальное имя для значения подвыражения `e1`. Снижать за это оценку не нужно.
- Стоит снизить оценку минимум на балл, если решение сложнее, чем приведенное выше.
- Следует снизить оценку на 1 балл, если в решении не используется `closureFun`, `closureEnv`, `funArg`, `funName` или `funBody`.
- Следует снизить оценку на 1 балл, если какая-то из функций `closureFun`, `closureEnv`, `funArg`, `funName` или `funBody` вызывается дважды от одного и того же аргумента.
- Ваша оценка должна быть беспристрастной. Вполне нормально, если решение отличается от приведенных выше вариантов. Вы проверяете правильность решения и его стиль, а не степень совпадения решения с приведенными вариантами.

Оценка для Критерий 17

5

Комментарий к Критерий 17

Критерий 18

Оценим реализацию вычисления значений `IntNum`, `Unit` и `Closure`.

Так как вычисление значения языка MUPL должно равняться самому значению, а других нерассмотренных выражений MUPL кроме этих не осталось, логичнее в конце определения функции `evalUnderEnv` добавить

```
evalUnderEnv e _ = e
```

и это будет правильной реализацией.

- Указанная строка кода должна быть последней в определении функции `evalUnderEnv`. Если это не так, то за решение следует поставить не более 1 балла.
- Стоит снизить оценку на два балла, если в определении `evalUnderEnv` есть любая из следующих строк (имена параметров могут отличаться)

```
evalUnderEnv (IntNum n) _ = IntNum n  
evalUnderEnv Unit _ = Unit  
evalUnderEnv (Closure env fun) _ = Closure env fun
```

- Ваша оценка должна быть беспристрастной. Вполне нормально, если решение отличается от приведенных выше вариантов. Вы проверяете правильность решения и его стиль, а не степень совпадения решения с приведенными вариантами.

Оценка для Критерий 18

5

Комментарий к Критерий 18

Критерий 19

Реализация `ifAUnit` должна иметь вид

```
ifAUnit e1 e2 e3 =  
  IfGreater (IsAUnit e1) (IntNum 0) e2 e3
```

- Решение заслуживает не более 1 балла, если в правой части равенства используются конструкции или функции Haskell.
- Следует снизить оценку на 2 балла, если реализация представлена более сложным решением.
- Ваша оценка должна быть беспристрастной. Вполне нормально, если решение отличается от приведенных выше вариантов. Вы проверяете правильность решения и его стиль, а не степень совпадения решения с приведенными вариантами.

Оценка для Критерий 19

5

Комментарий к Критерий 19

Критерий 20

Реализация `mLet` должна иметь вид

```
mLet [] e2 = e2  
mLet (x : xs) e2 = Let x (mLet xs e2)
```

- Решение заслуживает не более 1 балла, если в правой части равенства используются конструкции или функции Haskell, отличные от имен макросов MUPL.

- Следует снизить оценку на 2 балла, если реализация представлена более сложным решением.
- Ваша оценка должна быть беспристрастной. Вполне нормально, если решение отличается от приведенных выше вариантов. Вы проверяете правильность решения и его стиль, а не степень совпадения решения с приведенными вариантами.

Оценка для Критерий 20

5

Комментарий к Критерий 20

Критерий 21

Реализация `ifEq` должна иметь вид

```
ifEq e1 e2 e3 e4 =  
  mLet [("x", e1), ("y", e2)]  
    (IfGreater (Var "x") (Var "y")  
               e4  
               (IfGreater (Var "y") (Var "x")  
                           e4  
                           e3))
```

- Решение заслуживает не более 1 балла, если в правой части равенства используются конструкции или функции Haskell, отличные от имен макросов MUPL.
- Решение заслуживает не более 1 балла, если в правой части дважды упоминается `e1` или `e2`.
- Решение заслуживает не более 1 балла, если при каком-то варианте ее выполнения `e3` или `e4` будет вычислено дважды.
- Следует снизить оценку на 2 балла, если реализация представлена более сложным решением.
- Ваша оценка должна быть беспристрастной. Вполне нормально, если решение отличается от приведенных выше вариантов. Вы проверяете правильность решения и его стиль, а не степень совпадения решения с приведенными вариантами.

Оценка для Критерий 21

5

Комментарий к Критерий 21

Критерий 22

Реализация `mMap` может иметь вид

```
mMap =
  Fun "" "f"
    (Fun "loop" "l"
      (ifAUnit (Var "l")
        Unit
        (Pair (Call (Var "f")
                    (Head (Var "l"))))
          (Call (Var "loop")
                (Tail (Var "l"))))))))
```

или

```
mMap =
  Fun "map" "f"
    (Fun "" "l"
      (ifAUnit (Var "l")
        Unit
        (Pair (Call (Var "f")
                    (Head (Var "l"))))
          (Call (Call (Var "map") (Var "f"))
                (Tail (Var "l"))))))))
```

Оба определения являются корректными.

- Решение заслуживает не более 1 балла, если в правой части равенства используются конструкции или функции Haskell, отличные от имен макросов MUPL.
- Следует снизить оценку на 2 балла, если реализация представлена более сложным решением, чем представленные выше.
- Ваша оценка должна быть беспристрастной. Вполне нормально, если решение отличается от приведенных выше вариантов. Вы проверяете правильность решения и его стиль, а не степень совпадения решения с приведенными вариантами.

Оценка для Критерий 22

5

Комментарий к Критерий 22

Критерий 23

Реализация `mMapAddN` может иметь вид

```
mMapAddN =
  Call (Fun "" "x"
    (Call mMap
      (Fun "" "y"
        (Add (Var "x") (Var "y"))))))
```

- Решение заслуживает не более 1 балла, если в правой части равенства используются конструкции или функции Haskell, отличные от имен макросов MUPL.
- Следует снизить оценку на 2 балла, если реализация не использует `mMap`.
- Следует снизить оценку минимум на один балл, если решение сложнее, чем представленное выше.

- Ваша оценка должна быть беспристрастной. Вполне нормально, если решение отличается от приведенных выше вариантов. Вы проверяете правильность решения и его стиль, а не степень совпадения решения с приведенными вариантами.

Оценка для Критерий 23

5

Комментарий к Критерий 23

Критерий 24

Реализация `fact` может иметь вид

```
fact =
  mLet [ ("sub1", Fun "sub1" "x"
          (Call (Fun "loop" "y"
                  (Let ("z", Add (IntNum 1) (Var "y"))
                      (ifEq (Var "z") (Var "x")
                          (Var "y")
                          (Call (Var "loop") (Var "z")))))
                (IntNum 0)))
        , ("mult", Fun "mult" "x"
          (Fun "loop" "y"
            (ifEq (Var "y") (IntNum 0)
                (IntNum 0)
                (Add (Var "x")
                    (Call (Var "loop")
                        (Call (Var "sub1") (Var "y"))))))))
    ]
  (Fun "fact" "n"
    (IfGreater (IntNum 2) (Var "n")
      (IntNum 1)
      (Call (Call (Var "mult") (Var "n"))
        (Call (Var "fact")
          (Call (Var "sub1")
            (Var "n"))))))))
```

Здесь `sub1` и `mult` вспомогательные функции для уменьшения аргумента на единицу и для умножения соответственно.

- Решение заслуживает не более 1 балла, если в правой части равенства используются конструкции или функции Haskell, отличные от имен макросов MUPL.
- Решение заслуживает не более 1 балла, если вспомогательные функции описаны отдельными Haskell-овскими определениями.
- Решение заслуживает не более 1 балла, если в коде присутствует `IntNum` от отрицательного числа.
- Решение может быть представлено в несколько иной форме, без использования функции уменьшения аргумента на единицу.
- Решение может быть представлено в виде реализации с хвостовой рекурсией. Снижать за это оценку не нужно.
- Ваша оценка должна быть беспристрастной. Вполне нормально, если решение отличается от приведенных выше вариантов. Вы проверяете правильность решения и его стиль, а не степень совпадения решения с приведенными вариантами.

Оценка для Критерий 24

5

Комментарий к Критерий 24

НАВИГАЦИЯ



В начало

- Личный кабинет


Страницы сайта

Мои курсы


Графика Осень 2018

ЯП 2018

Участники

 Значки

 Компетенции

 Оценки

Общее

Форумы курса

Материалы по тематике курса

Лабораторные работы

Оценка лабораторных работ

 Лабораторная работа №0

 Лабораторная работа №1

 Лабораторная работа №2

 Лабораторная работа №3

 **Лабораторная работа №4**

- Моя работа

 Лабораторная работа №5

 Лабораторная работа №6

Раздел 1. Standard ML

Раздел 2. Haskell

Раздел 3. LISP

Раздел 4. Ruby

Раздел 5. PROLOG

Реферат

Аттестация

Бонусы

Напоминалки

ИКБ

On-line

