

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Сыров Игорь Анатольевич  
Должность: Директор  
Дата подписания: 30.10.2023 11:00:29  
Уникальный программный ключ:  
b683afe664d7e9f64175886cf9626a196149ad36

СТЕРЛИТАМАКСКИЙ ФИЛИАЛ  
ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО ОБРАЗОВАТЕЛЬНОГО  
УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ  
«УФИМСКИЙ УНИВЕРСИТЕТ НАУКИ И ТЕХНОЛОГИЙ»

Факультет Математики и информационных технологий  
Кафедра Прикладной информатики и программирования

**Оценочные материалы по дисциплине (модулю)**

дисциплина **Функциональное программирование**

**Блок Б1, часть, формируемая участниками образовательных отношений,  
Б1.В.ДВ.02.01**

цикл дисциплины и его часть (обязательная часть или часть, формируемая участниками образовательных отношений)

Направление

**02.03.03 Математическое обеспечение и администрирование информационных систем**

код наименование направления

Программа

**Сетевое программирование и администрирование информационных систем**

Форма обучения

**Очная**

Для поступивших на обучение в  
**2023 г.**

Разработчик (составитель)

**д.ф.-м.н., профессор**

**Хусаинов И. Г.**

ученая степень, должность, ФИО

<b>1. Перечень компетенций, индикаторов достижения компетенций и описание показателей и критериев оценивания результатов обучения по дисциплине (модулю)</b> .....	<b>3</b>
<b>2. Оценочные средства, необходимые для оценки результатов обучения по дисциплине (модулю)</b> .....	<b>5</b>
<b>3. Методические материалы, определяющие процедуры оценивания результатов обучения по дисциплине (модулю), описание шкал оценивания</b> .....	<b>36</b>

**1. Перечень компетенций, индикаторов достижения компетенций и описание показателей и критериев оценивания результатов обучения по дисциплине (модулю)**

Формируемая компетенция (с указанием кода)	Код и наименование индикатора достижения компетенции	Результаты обучения по дисциплине (модулю)	Показатели и критерии оценивания результатов обучения по дисциплине (модулю)				Вид оценочного средства
			1	2	3	4	
			неуд.	удовл.	хорошо	отлично	
ПК-3. Способен использовать основные концептуальные положения функционального, логического, объектно-ориентированного и визуального направлений программирования, методы, способы и средства разработки программ в рамках этих направлений	ПК-3.1. Знает основные концептуальные положения функционального, логического, объектно-ориентированного и визуального направлений программирования, методы, способы и средства разработки программ в рамках этих направлений.	Обучающийся должен: знать особенностей функционального подхода в разработке программного обеспечения.	Обучающийся не знает основные операторы, функции и типы функционального языка программирования Haskell.	Обучающийся плохо знает основные операторы, функции и типы функционального языка программирования Haskell.	Обучающийся частично знает основные операторы, функции и типы функционального языка программирования Haskell.	Обучающийся хорошо знает основные операторы, функции и типы функционального языка программирования Haskell.	Устный опрос.
	ПК-3.2. Умеет программировать в рамках этих	Обучающийся должен: уметь использовать	Обучающийся не умеет применять	Плохо умеет применять операторы и	Частично умеет применять операторы и	Хорошо умеет применять операторы и	Тестовые задания.

	направлений.	функциональные методы при решении простых вычислительных задач.	операторы и стандартные функции функционального языка Haskell при составлении программ.	стандартные функции функционального языка Haskell при составлении программ.	стандартные функции функционального языка Haskell при составлении программ.	стандартные функции функционального языка Haskell при составлении программ.	
	ПК-3.3. Имеет практический опыт разработки программ в рамках этих направлений.	Обучающийся должен: владеть методами использования в профессиональной деятельности функционального языка программирования.	Обучающийся не владеет методами использования в профессиональной деятельности функционального языка программирования.	Обучающийся слабо владеет методами использования в профессиональной деятельности функционального языка программирования.	Обучающийся частично владеет методами использования в профессиональной деятельности функционального языка программирования.	Обучающийся хорошо владеет методами использования в профессиональной деятельности функционального языка программирования.	Контрольная работа, лабораторная работа.

**2. Оценочные средства, необходимые для оценки результатов обучения по дисциплине (модулю)**

**Перечень вопросов для устного опроса**

**Модуль №1**

1. Основные свойства функциональных языков.
2. Когда была предложена первая версия языка Haskell?
3. Основные характеристики языка Haskell.
4. Что такое краткость языка Haskell?
5. Что такое простота языка Haskell?
6. Что такое строгая типизация?
7. Что такое модульность?

**Модуль №2**

8. Что означает «функции - это значения»?
9. В чем заключается чистота языка?
10. Что означает параллелизм?
11. Каким образом выполняются отложенные вычисления?
12. Как происходит оптимизация?
13. Бесконечные структуры данных?
14. Функции для работы с кортежами.
15. Функции для работы со списками.
16. Допустимые имена переменных и функций.
17. Условные выражения в языке Haskell.
18. Определение функций в языке Haskell.
19. Правила выравнивания.
20. Сопоставление с образцом.
21. Операция выбора.
22. Кусочное задание функций.
23. Функции высших порядков.

**Задания для компьютерного тестирования**

К какому классу относится язык программирования Haskell ?,

Функциональный,

Структурный,

Язык параллельного программирования,

Сценарный,

Выберите правильный вариант систем типов реализованных в Haskell ,

полная, сильная, статическая, с выводом типов ,

статическая, слабая, с выводом типов ,

динамическая, сильная, с выводом типов,

с выводом типов, статическая, полная, слабая ,

Выберите характеристику которая не относится к языку Haskell ,

праильные выражения,

функции высшего порядка,

частичное применение,  
параметрический полиморфизм,  
Функция, принимающая в качестве аргументов другие функции или  
возвращающая другую функцию в качестве результата - это,  
функции высшего порядка,  
ленивые вычисления,  
генераторы списков ,  
Функциональные зависимости ,

Преобразование функции от многих аргументов в функцию, берущую  
свои аргументы по одному - это ...,

частичное применение,  
ленивые вычисления,  
Недопустимость побочных эффектов ,  
статическая сильная полная типизация,  
let neg x = if x = false then true else false является примером для ... ,  
сопоставление с образцом,  
параметрический полиморфизм,  
алгебраические типы данных,  
генераторы списков ,

Полиморфизм ранга k (при  $k > 1$ ) позволяет конкретизировать  
переменные типа полиморфными типами ранга не выше,

k - 1,  
k,  
k+1,  
2\*k,

Сколько конструкторов имеет список ?,

2,  
1,  
3,  
4,

Для формализации и анализа понятия вычислимости используется ...,

лямбда-исчисление,  
гамма-исчисление,  
бетта-исчисление,  
сигма-исчисление ,

Параметрическому полиморфизму традиционно противопоставляется

...,

ad hoc полиморфизм,  
lc hoc полиморфизм,  
ul hoc полиморфизм,  
ab hoc полиморфизм,

Выделение этой промежуточной категории важно в свете того, что  
выведение типов для Системы F, типизированной по Карри, разрешимо для k

...,

$\leq 2$ ,

>1,  
>=1,  
>0,

Полиморфизм высших рангов позволяет ставить кванторы переменных типа ... от произвольного числа стрелок в типах высших порядков,  
слева,  
справа,  
не имеет значения,  
в центре,

Метод анализа и обработки структур данных, основанный на выполнении определённых инструкций в зависимости от совпадения исследуемого значения с тем или иным образцом, в качестве которого может использоваться константа, предикат, тип данных или иная подде

Сопоставление с образцом,  
функциональные образцы,  
ленивые вычисления,  
частичное применение,

Простым примером алгебраического типа данных является ...,  
список,  
дек,  
стек,  
линейный список,

Какая из представленных характеристик соответствует утверждению "Программы, написанные на чистых языках программирования, проще отлаживать, верифицировать, в них проще обнаруживать ошибки, которые не удалось обнаружить с помощью тестирования"?,

чистота языка,  
охраняющие выражения,  
сопоставление с образцом,  
функциональные образцы,

Стратегия вычисления, согласно которой вычисления следует откладывать до тех пор, пока не понадобится их результат - это ...,

lazy evaluation,  
pattern matching,  
guards,  
list comprehensions,

В каком году появился язык Haskell ?,  
1990,  
1992,  
1989,  
1993,

Статическая сильная полная типизация с автоматическим выводением типов, основанная на типизации ...,

Хиндли - Милнера,  
Роуена - Скота,

Пьера - Милнера,  
Скота - Хиндли,  
Полиморфизм ранга 1 называется ...,  
пренексным,  
первым,  
одномерным,  
одиночным,

Логическое выражение, которое предназначено для ограничения  
вычислительных процессов и выбора варианта вычислений - это ...,  
охраняющее выражение ,  
сопоставление с образцом,  
функциональные образцы,  
частичное применение,

Язык Haskell - это ...,  
стандартизированный чистый функциональный язык  
программирования общего назначения,  
язык программирования общего назначения,  
компилируемый статически типизированный язык  
программирования ,  
императивный, структурированный, объектно-ориентированный  
язык программирования,

Выберите основные характеристики языка Haskell.,  
все ответы верны,  
возможность использования лямбда-абстракции,  
функции высшего порядка,  
частичное применение,

Что такое типизация в Haskell?,  
связывание некоторого значения с его типом,  
разработка типовых конструкций или технологических  
процессов,  
один из способов создания образов,  
выделение существенного,

С помощью какого объявления мы можем определить собственные  
типы в Haskell?,

data,  
bool,  
false,  
case,

Выберите правильный вариант.,  
[1,3..] => [1,3,5,7,9, ...],  
[1..10] => [2,4,6,8,10],  
[1,3..] => [1,3,5,7,9],  
все ответы верны,

Что означает стрелка a->b?,  
функцию,



процесс,  
элемент ,  
нет правильного варианта,  
Что такое кортеж?,  
некоторый набор значений заключенный в () и разделенных  
запятыми,  
элемент отношения,  
совокупность взаимосвязанных данных,  
нет правильного варианта,  
Из чего состоит список?,  
только из элементов одного типа,  
из множеств,  
из четырех основных элементов: символов, слов, словосочетаний  
и предложений,  
все ответы верны,  
Охраняющие выражения...,  
логическое выражение, которое накладывает некоторые  
ограничения на используемые образцы,  
конструкция на языке программирования, предназначенная для  
выполнения вычислений,  
это язык текстовых шаблонов,  
вычисление значений функции,  
Выберите один из двух способов описания локальных функций,  
where,  
case ,  
reverse,  
data,  
Выберите правильный вариант.,  
if <условие> then <выражение1> else <выражение2>,  
if <выражение1> then <выражение1> else <условие>,  
if <выражение1> then <условие> else <выражение2> ,  
if <условие> else <выражение2>,  
Выберите правильный вариант.,  
reverse - разворачивание списка в обратном порядке,  
product - взять из списка n первых элементов,  
length - конкатенация списков,  
tail - элемент с заданным номером,  
Компиляторы и интерпретаторы языка Haskell работают с файлами с  
расширением ,  
\*.hs,  
\*.shs,  
\*.hss,  
\*hs,  
Аккумулятор - это ...,

дополнительный аргумент функции в котором накапливаются  
результаты вычислений,  
источник тока многоразового действия,  
конструкции, записанные в более близком к естественному языку  
виде,

переход по меткам,  
Арифметические последовательности создаются...,  
с помощью образцов,  
с помощью непустых списков,  
с помощью конструктора,  
нет правильного варианта,

Для чего используется замыкание списков?,  
для создания нового списка из существующего списка по  
некоторым условиям,  
для создания двух списков из существующего списка ,  
для удаления списка из существующего списка по некоторым  
условиям,

нет правильного варианта,  
null - это?,  
проверка списка на пустоту,  
логическое отрицание,  
макрос, объявленный в заголовочном файле,  
вычисляет пустых указатели,

Выберите правильный вариант.,  
функции могут принимать и возвращать значения, имеющие  
строго тот же тип данных, что указан при описании функции,  
каждая операция не требует аргументов строго определённых  
типов,

допускается явное преобразование типов,  
аргументы привязываются к разным типам,

Укажите правильные варианты описания case,  
case <выражение0> of <образец1> -> <выражение1> <образец2> -  
> <выражение2> ... <образецN> -> <выражениеN>,  
case <образец1> -> <выражение1> <образец2> -> <выражение2>  
... <образецN> -> <выражениеN>,  
case <выражение0> of <образец1> -> <выражениеN>,  
нет правильного варианта,

Как вычисляется факториал?,  
 $f\ 0 = 1$   $f\ n = n * f(n-1)$  ,  
 $f\ 0 = 1$   $f\ n = n * f(n)$  ,  
 $f\ n = f$   $f\ n = n * f(n-1)$  ,  
 $f\ n = n * f(n-1)$  ,

(x:xs) конструкция единственным образом раскладывает список на,  
голову и хвост,  
голову и тело,

хвосту и телу,  
 нет верного ответа,  
 какая функция с условием на список проверяет принадлежность  
 элемента к списку,  
 elem::a->[a]->Bool,  
 any::Bool->[a]->Bool,  
 null::[a]->Bool,  
 filter::Bool->[a]->[a],  
 значения, которые могут быть переданы другим в качестве аргумента  
 или возвращать в качестве результата,  
 функция,  
 кортеж,  
 список,  
 аккумулятор,  
 какая функция возвращает все элементы кроме последнего,  
 init::[a]->a ,  
 last::[a]->a,  
 init::[a]->[],  
 take::Int->[a]->[a],  
 Какая связка операторов дает выбор из 2-х вариантов?,  
 if:then:else,  
 For:to:do,  
 Case:of: end,  
 Begin:end,  
 Какая связка операторов не является циклом?,  
 if:then:else,  
 For:to:do,  
 repeat:until,  
 While:do,  
 Какая из операций определяет остаток целочисленного деления?,  
 mod,  
 div,  
 in,  
 or,  
 Какая из операций является логической?,  
 or,  
 div,  
 mod,  
 in,  
 Функция отбрасывающая n элементов из начала списка,  
 drop::Int->[a],  
 take::Int->[a]->[a],  
 reverse::[a]->[a],  
 нет верного варианта,

дополнительный аргумент функции в котором накапливаются  
результаты вычисления,  
аккумулятор,  
список,  
охраняющее выражение,  
нет верного варианта,

Для вычисления квадратного корня применяется функция:,  
SQRT(X),  
SQR(X),  
ORD(X),  
EXP(X),

Какая функция переводит вещественную переменную в строковую,  
Int To Str(k) ,  
Float To Str(k),  
Str To Int (k),  
Str To Float(k),

Какая функция переводит строковую переменную в целую,  
Str To Int (k) ,  
Float To Str(k),  
Int To Str(k),  
Str To Float(k),

Какая функция переводит строковую переменную в вещественную,  
Str To Float ,  
Float To Str(k),  
Int To Str(k),  
Str To Int(k),

Какие из перечисленных множеств являются счетными?,  
натуральный ряд,  
множество действительных чисел,  
конечное множество,  
нет верного ответа,

образцами не могут быть,  
кортеж,  
константа,  
имя,  
знак подчеркивания,

\*\* что делает эта функция,  
возвращает число в вещественную степень,  
находит остаток,  
целочисленная степень,  
целочисленное деление,

функция возвращает числа в целочисленную положительную степень,  
^^,  
\*\*,  
/,

\*,  
набор значений заключенные в круглые скобки и разделенные  
запятыми,

кортеж,  
список,  
комментарий,  
идентификатор,

Какой тип из перечисленных занимает в памяти ровно 1 байт?,

Char,  
Integer,  
Word,  
Longint,

Haskell это:

функциональный декларативный язык программирования;,  
математический язык программирования;,  
логический язык программирования;,  
объектно-ориентированный язык программирования;,

Какие бывают конструкции ветвления?,

if, case и охраняющие выражения;,  
if и case;,  
if, while и for;,  
if и охраняющие выражения;,

Конструкция ветвления if: ,

If <условие> then <выражение1> else <выражение2>;,  
If <условие> then <выражение1>;,  
If <условие> then <выражение1> <выражение2>;,  
If <условие> else <выражение>;,

<выражение1> и <выражение2> в конструкции if обязательно должны  
быть:;

одного типа;,  
разных типов;,  
типа Bool;,  
функциями;,

<условие> в конструкции if обязательно должно быть:;

типа Bool;,  
функцией;,  
типа Char;,  
типа Integer;,

Конструкция ветвления case:;

case <выражение 0> of под ним с отступом <образец n> -> <оператор  
n>;,

case <выражение 0> of <образец n> -> <оператор n>;,  
case <выражение 0> of -> <оператор n>;,  
case <выражение 0> of <образец n>;,

Каждый <образец n> должен быть:;

написан на новой строке и с отступом от case (использование  
 двумерного синтаксиса);,  
 написан на новой строке;,  
 типа Bool;,  
 функцией;,  
 Охранные выражения - это:,  
 логические выражения, которые накладывают некоторые ограничения  
 на используемые образцы;,  
 логические выражения, которые ограничивают количество условий;,  
 функции, которые будут выполнены при определенных условиях;,  
 образцы, которые ограничивают условие;,  
 Конструкция охранного выражения:,  
 <имя функции> <список аргументов> под ним с отступом | <условие  
 n> = <выражение n>;,  
 <имя функции> <список аргументов>| <условие 1> = <выражение 1>;,  
 <имя функции> <список аргументов>| = <выражение 1>;,  
 <имя функции>| = <выражение 1>;,  
 При вычислении значения функции в охранных выражениях  
 происходит:,  
 сопоставление с образцами;,  
 сопоставление с ограничениями;,  
 сравнение с числом;,  
 сравнение с функцией;,  
 Что поддерживает Haskell? ,  
 отложенные вычисления;,  
 объекты;,  
 графики;,  
 таблицы;,  
 В Haskell есть: ,  
 встроенная поддержка многозадачного и параллельного  
 программирования;,  
 графики;,  
 таблицы;,  
 встроенная поддержка виртуальных объектов;,  
 Что делает следующая программа?  $f\ x\ y = \text{if } x > y \text{ then } x \text{ else } y$  ,  
 находит максимум между  $x$  и  $y$ ;,  
 находит минимум между  $x$  и  $y$ ;,  
 находит разность между  $x$  и  $y$ ;,  
 находит сумму между  $x$  и  $y$ ;,  
 Просмотр выражений в охраняющих выражениях происходит: ,  
 сверху вниз до первого выражения, значение которого = true;,  
 снизу вверх до первого выражения, значение которого = true;,  
 сверху вниз до первого выражения, значение которого = false;,  
 снизу вверх до первого выражения, значение которого = false;,  
 Otherwise в охранных выражениях используется: ,

если никакое иное выражение не было истинно;,  
если все выражения были истинны;,  
если нет других выражений;,  
если все выражения равны 0;,

В честь кого был назван язык программирования Haskell?,  
в честь математика Хаскелла Карри;,  
в честь физика Хаскелла Карри;,  
в честь математика Роберта Хаскелла; ,  
в честь физика Роберта Хаскелла;,

Сколько может быть образцов в конструкции case?,  
n образцов;,  
(n-1) образцов;,  
2 образца; ,  
1 образец;,

Что делает следующая программа? `case n of 0->"zero" 1->"one" ... 9->"nine"`,

при вводе с клавиатуры цифры, выдается его английская запись;,  
выводит цифры при вводе их английских записей;,  
выводит цифры;,  
выводит обратный порядок цифр;,

Оператор `if` предназначен для;,  
ветвления вычислительного процесса в зависимости от условия булевского типа;,

множественного ветвления, когда вычислительный процесс разбивается на несколько ветвей в зависимости от значения выражения произвольного типа;,

ветвления вычислительного процесса в зависимости от условия произвольного типа;,

множественного ветвления, когда вычислительный процесс разбивается на несколько ветвей в зависимости от значения выражения булевского типа;,

Оператор `case` предназначен для;,  
множественного ветвления, когда вычислительный процесс разбивается на несколько ветвей в зависимости от значения выражения произвольного типа;,

ветвления вычислительного процесса в зависимости от условия булевского типа;,

ветвления вычислительного процесса в зависимости от условия произвольного типа;,

множественного ветвления, когда вычислительный процесс разбивается на несколько ветвей в зависимости от значения выражения булевского типа;,

Функциональное программирование относится к,  
декларативному программированию,  
императивному программированию,

объектно-ориентированному программированию,  
 нет верных вариантов,  
 Декларативный подход в программировании требует,  
 описывать свойства требуемого результата,  
 описывать последовательность действий для нахождения  
 результата,  
 нет верных вариантов,  
 описывать последовательность действий и свойства требуемого  
 результата,  
 К особенностям декларативных языков программирования относятся,  
 выразительность, параллелизм и ориентированность на присущий  
 человеку образ мышления,  
 параллелизм, полиморфизм и наследование,  
 выразительность, инкапсуляция и ориентированность на  
 присущий человеку образ мышления,  
 нет верных вариантов,  
 К языкам функционального программирования относятся,  
 Lisp, Haskell, Prolog, OCaml, F#, Clean,  
 Lisp, Scheme, Miranda, Haskell, OCaml, F#, Erlang,  
 Lisp, Haskell, F#, C#, C++, Object Pascal,  
 Lisp, Miranda, Haskell, JavaScript, Visual Prolog, OCaml, F#, Erlang,  
 К языкам логического программирования относятся,  
 Prolog, F#,  
 Haskell, Cofer,  
 Prolog, Visual Prolog,  
 нет верных вариантов,  
 Язык функционального программирования F# наследует от языков,  
 OCaml, Haskell,  
 Lisp, OCaml,  
 Lisp, Haskell,  
 нет верных вариантов,  
 Тип выражения говорит о том,  
 какие значения могут получиться при вычислении этого  
 выражения и получатся ли какие-нибудь значения вообще,  
 какие значения могут получиться при вычислении этого  
 выражения,  
 имеет ли выражение какое-нибудь значение или нет,  
 нет верных вариантов,  
 Базовый элемент синтаксиса F#,  
 выражение,  
 константа,  
 объект,  
 класс,  
 Функции в F# объявляются конструкцией,  
 let <имя функции> (<аргумент функции>) = <тело функции>,



<имя функции> (<аргумент функции>) := <тело функции>,  
let <тип функции> <имя функции> (<аргумент функции>) :  
{<тело функции>},

нет верных вариантов,  
Функция let plus x y = x+y;; будет иметь тип,  
int -> int -> int,  
int -> int,  
нет верных вариантов,  
int\*int -> int,

Результат выражения 0.1+1 будет,  
ошибка,  
1.1,  
1,  
0.1,

В классических функциональных языках многомерные массивы представляются,

списками списков,  
списками массивов,  
массивами списков,  
нет верных вариантов,

Функция конкатенации (<склейки>) списков [1;2;3] и [4;5] будет записана как,

[1;2;3] @ [4;5],  
[1;2;3] + [4;5],  
[1;2;3][4;5],  
[1;2;3] and [4;5],

Хвостом списка называется,  
все элементы списка кроме первого,  
первый элемент списка,  
последний элемент списка,  
нет верного варианта,

Лямбда-выражение ?x.x<sup>2</sup>-x в F# будет записано в виде,

fun x -> x\*x-x,  
let f x = x\*x-x,  
нет верных вариантов,  
let function x -> x\*x-x,

Программная секция, содержащая описание запросов - это,

секция целей (goal),  
секция доменов (domains),  
секция предикатов (predicates),  
секция утверждений (clauses),

Запись в Haskell f (y x) обозначает запись в математике,

f(y(x)),  
f(y,x),  
f(y,x),

$f(y) + f(x)$ ,

Запись в Haskell  $f (g x) (h y)$  обозначает запись в математике,

$f(g(x),h(y))$ ,

$f(g,x) + f(h,y)$ ,

$f (g,x,h,y)$ ,

$f(x) + f(y) + f(g) + f(h)$ ,

Вызов функции от отрицательного аргумента записывается,

$f (-x)$ ,

$f -x$ ,

$f --x$ ,

$f x$ ,

При вызове функции по значению ,

ее аргументы сначала вычисляются, а потом в тело функции передаются полученные значения,

в тело функции передаются полученные значения, а потом ее аргументы вычисляются,

значения и аргументы вычисляются параллельно,

Haskell не допускает вызов функции по значению,

Функция называется строгой по одному из своих аргументов, если, всегда требует его значение для получения результата,

указать в типе переменной константу,

этот аргумент переопределен в функции,

один из аргументов имеет тип список,

Функциональное программирование черпает идеи из ,

комбинаторной логики и лямбда-исчисления,

высшей математики и геометрии Лобачевского,

нелинейной алгебры,

математической физики,

Одними из первых языков с функциональным стилем был,

LISP,

APL,

ISWIM,

FP,

Условия стоят ,

после знака | и до знака =,

между знаками | и =,

нет определенного правила где ставить условие,

после =,

Если требуется создать локальные привязки, то используется,

let,

if,

case,

лямбда функции,

Если необходимо применить функцию к результату вычисления другой, то следует записать как,

$(f.g) x = f (g x),$

$(f.g) = (g x) f,$

$(f.g) f = (g.x),$

$(f g) = (g.x) x,$

Пример анонимной функции,

$\backslash x y \rightarrow \text{sqrt} (x*x + y*y),$

$\backslash x \backslash y \rightarrow \text{sqrt} (x*x + y*y),$

$\backslash x y = \text{sqrt} (x*x + y*y),$

$\backslash x \backslash y = \text{sqrt} (x*x + y*y),$

Запись  $(f.g)$  означает что ,

сначала к аргументу применяется  $g$ , а затем к результату применяется  $f$ ,

сначала к аргументу применяется  $f$ , а затем к результату применяется  $g$ ,

Применяться параллельно,

Нет верного ответа,

Проверка на пустой список,

`null`,

`tail`,

`length`,

`take`,

Выбор головы в непустом списке,

`head`,

`reverse`,

`sum`,

`zip`,

Объединение списка списков,

`concat`,

`tail`,

`reverse`,

`take`,

Объединение двух списков 16 ,

`++`,

`sum`,

`!!`,

`product`,

Объединение двух списков в список пар,

`zip`,

`concat`,

`sum`,

`head`,

Удаление первых  $n$  элементов,

`drop`,

`product`,

`concat`,

`null`,

Выбор первых  $n$  элементов,

take,  
drop,  
product,  
zip,  
Доступ по индексу,  
!!,  
take,  
head,  
concat,  
Список - это ...,  
абстрактный тип данных, представляющий собой упорядоченный  
набор значений, в котором некоторое значение может встречаться более  
одного раза,  
целочисленный тип данных,  
булевый тип данных,  
один из механизмов функционального программирования  
предназначенный для оптимизации определения функций,  
Картеж - это ...,  
некоторый набор значений, заключенных в () и разделённый ",",  
являющимся частным случаем списка,  
общий случай списка,  
вещественный тип данных,  
функция используемая для работы со списками,  
Результат выполнения  $f(x:xs) = x$  ,  
первый элемент списка,  
последний элемент списка,  
весь список,  
пустой список,  
Результат выполнения `Main> f [3,5,5,9]` для программы:  $f(x:xs) = -x$ ,  
-3,  
9,  
-9,  
3,  
Каким языком является Haskell?,  
право ассоциативным,  
лево ассоциативным,  
неассоциативным,  
правоассоциативным и левоассоциативным,  
Какой из списков является бесконечным?,  
[1,2..],  
[],  
[1..100],  
(x:xs),  
Название специального модуля, используемого для работы со  
списками?,

Data.List,  
 Prelude,  
 IO,  
 Main,  
 Какая функция вернет обратный список?,  
 reverse,  
 !!,  
 tail,  
 take,  
 Результат выполнения Main>f [3,7,-5,9] для программы: fa = reversea,  
 [9,-5,7,3],  
 [3],  
 [9,-5,7],  
 [9],  
 Команда splitAt n xs это ...,  
 команда, которая вернет пару списков, полученных из списка xs  
 разбиением с n-й позиции (причем n-й элемент войдет во второй список),  
 команда, которая вернет список путем слияния двух списков,  
 команда, которая возвращает все элементы списка кроме последнего,  
 команда, которая вернет пару списков, полученных из списка xs  
 разбиением с n-й позиции (причем n-й элемент войдет в первый список),  
 Замыкание используется для ...,  
 создания нового списка по некоторому условию,  
 удаления списка,  
 нахождения элемента под номером,  
 слияния двух списков,  
 Какой тип возвращает команда null,  
 Bool,  
 Int,  
 [a],  
 [Bool],  
 findIndex p xs выполняет?,  
 поиск и возвращает индекс первого элемента в списке xs (обернутый  
 конструктором Just), который удовлетворяет предикату p или Nothing в  
 противном случае,  
 поиск и возвращает первый элемент в списке xs (обернутый  
 конструктором Just), который удовлетворяет предикату p или Nothing в  
 противном случае ,  
 осуществляет сортировку списка xs, используя специальным образом  
 заданную программистом функцию p,  
 ни чего из перечисленного,  
 Пустой список это?,  
 список без элементов,  
 список, в котором все элементы равны null,  
 список, в котором есть один элемент,

вовсе не список,  
 Какой список будет из четных положительных элементов?,  
 $[ 2*x \mid x \leftarrow [1..30]]$  ,  
 $[ x \mid x \leftarrow [1..30]]$  ,  
 $[ 2*x \mid x \leftarrow [-30..30]]$ ,  
 $[ 1,3..]$ ,  
 Какая функция используется для сортировки?,  
 sortBy,  
 find,  
 filter,  
 length,  
 Какая из функций является базовой для работы со списками?,  
 tail,  
 nub,  
 find,  
 delete,  
 Какая из функций не является базовой для работы со списками?,  
 delete,  
 head,  
 init,  
 reverse,  
 Какие команды используются для изменения списка или его  
 элементов?,  
 map zip,  
 find tail,  
 isPrefixOf, isSuffixOf, isInfixOf,  
 nullelem,  
 функция, которая может принимать на вход функции или возвращать  
 функции в качестве результата.,  
 Функция высшего порядка ,  
 Функция низшего порядка ,  
 Функция первого порядка ,  
 Функция второго порядка ,  
 Какая функция ничего не делает с аргументом, а просто возвращает  
 его?,  
 функция тождества,  
 константная функция,  
 функция композиции,  
 функция перестановки,  
 Какая функция заменит в списке каждый конструктор (:) на функцию  
 композиции, а пустой список на функцию id?,  
 foldr,  
 foldNat,  
 const,  
 flip,

Каким символом обозначается функция применения?,

\$,

@,

%,

::,

С помощью какой команды можно узнать в интерпретаторе приоритет функции?,

:i,

:d,

:h,

:n,

Какая функция имеет высший приоритет и очень часто используется при определении функции в бесточечном стиле? ,

аналогия с числами ,

функция перестановки ,

приоритет функции композиции ,

функция перестановки,

Сколько функций принимает функция композиции?,

2,

3,

4,

1,

Функция (Succ. ) - это \_\_\_\_\_ сечение операции (.),

левое,

правое,

нижнее,

верхнее,

Какая функция перед применением бинарной функции пропускает аргументы через унарную функцию?,

on,

flip,

foldr,

id,

Что означает фраза "большой приоритет?",

сначала умножение потом сложение,

сначала сложение потом умножение,

сначала умножение потом вычитание,

сначала сложение потом вычитание,

Какая функция принимает функцию двух аргументов и меняет аргументы местами?,

перестановки,

тождества,

композиции,

применения,

Какое значение выведет на экран следующая функция: Prelude> foldr (-)  
0 [1,2,3,4]?,

-2,

-1,

3,

4,

Из каких двух понятий складывается приоритет функций в Haskell?,

старшинство и ассоциативность,

меньшинство и ассоциативность,

старшинство и диссоциативность,

старшинство и меньшинство,

Операции (+), (-) и (\*) являются,

лево-ассоциативными,

право-ассоциативными,

лево-диссоциативными,

право-диссоциативными,

С помощью чего реализованы функции, возвращающие несколько значений?,

кортежей,

строк,

переменных,

идентификатора,

Что такое списки?,

полимерная структура данных,

сформированная структура данных,

это очень удобное средство, которое показывает текстовую

информацию,

начинающиеся с маркера или цифры взаимосвязанные между собой

предложения,

Как записывается список?,

[..],

{..},

(..),

/../,

Для чего используется функция head xs?,

вернет первый элемент списка,

вернет последний элемент списка,

удалит первый элемент списка,

вернет обратный список,

Для чего используется функция length xs.....,

вернет длину списка ,

вернет обратный список,

вернет список без первого элемента,

вернет список без последнего элемента,

Функция xs++[x] предназначена для....,



добавления x в качестве последнего элемента к списку xs,  
 добавления x в качестве первого элемента к списку xs,  
 добавления xs в качестве последнего элемента к списку x,  
 добавления xs в качестве первого элемента к списку x,  
 Take n xs используется для...,  
 возвращения списка из первых n элементов из списка xs,  
 возвращения списка из первых xs элементов из списка n,  
 удаление списка из первых n элементов из списка xs,  
 удаление списка из первых xs элементов из списка n,  
 Какие команды не входят в функции с условиями?,  
 drop,  
 null,  
 any test,  
 filter test,  
 Команда map используется для...,  
 применения функции ко всем элементам из списка и вернет в новый список,  
 применит функцию только для определенного элемента и добавит его в новый список,  
 вернет список в первоначальный вид,  
 нумерация элементов списка,  
 Какие функции всегда быстры?,  
 tail,  
 zip,  
 map,  
 last,  
 Какая команда становится медлительней от увеличения длины списка?,  
 last,  
 map,  
 zip,  
 tail,  
 Как записывается множество?,  
 {..},  
 [..],  
 (..),  
 /../,  
 Как правильно записать список всех натуральных чисел?,  
 [1..],  
 [1,3..],  
 [2,4..],  
 [1,3,..,99],  
 Какой модуль является базовым?,  
 Prelude,  
 Data,  
 List,

Data List,

Для чего используется функция import?,

для импортирования,

для переименования,

для удаления,

для копирования,

Функция nub ..,

возвращает список, в котором удалены повторяющиеся элементы,

возвращает список, в котором удалены неповторяющиеся элементы,

возвращает список, в котором удалены все элементы,

возвращает список, в котором удалены нужный элемент,

Функция sortBy используется для...,

сортировки списка,

сортировки элементов в списке,

сортировки всех списков,

сортировки выбранных элементов,

Функция union xs ys..,

возвращает объединение двух списков,

возвращает элементы двух списков,

возвращает повторяющиеся элементы двух списков,

возвращает неповторяющиеся элементы двух списков,

Функция findIndex...,

возвращает индекс первого элемента,

возвращает первый элемент,

возвращает индекс последнего элемента,

возвращает последний элемент,

Какая команда определяется как хвостовая рекурсия,

length,

for,

map,

map,

При трансляции компилятором преобразования используется ,

for,

while,

last,

head,

Haskell относится к ... языкам программирования,

ленивым,

быстрым,

сложным,

новым,

К особенностям языка можно отнести:,

все ответы верны,

алгебраические типы данных,

типы данных с параметрами,

рекурсивные типы данных,  
Возможно ли использование функций высшего порядка?,  
да,  
нет,  
частично,  
не знаю,  
Средства автоматизации разработки программ - это,  
инструментарий для системных аналитиков, разработчиков и программистов, позволяющий автоматизировать процесс проектирования и разработки программного обеспечения,  
по для системных аналитиков, разработчиков и программистов, позволяющее автоматизировать процесс проектирования и разработки программ,  
инструментарий для системных аналитиков, разработчиков и программистов, позволяющий автоматизировать процесс решения математических задач,  
инструментарий для программистов, позволяющий решать сложные вопросы разработки приложений,  
К компонентам интегрированного CASE-средства относятся:,  
все ответы верны,  
графические средства анализа и проектирования,  
репозиторий,  
средства тестирования,  
К компонентам интегрированного CASE-средства не относится:,  
средства копирования ,  
средства документирования,  
средства управления требованиями,  
средства реверсного инжиниринга ПО и баз данных,  
Выберите верное утверждение:,  
В Haskell есть возможность объединять модули в пакеты (package).,  
В Haskell отсутствует возможность объединять модули в пакеты (package).,  
В Haskell есть возможность объединять пакеты в модули.,  
В Haskell есть возможность объединять библиотеки в модули.,  
Могут ли пакеты зависеть от других пакетов?,  
да, если они пользуются модулями этих пакетов,  
да, если они не пользуются модулями этих пакетов,  
нет,  
да, если они пользуются разными модулями,  
Какое приложение существует для создания и установки пакетов?,  
cabal,  
labac,  
cab,  
cabl,  
Выберите верный вариант для определения свойства:,

ИмяСвойства : Значение,  
Значение:ИмяСвойства,  
ИмяСвойства::Значение,  
ИмяСвойства,Значение,  
Какое слово следует поместить за набором атрибутов, если пакет  
будет библиотекой?,  
Library,  
Executable,  
Class,  
LPakegs,  
Какое слово следует поместить за набором атрибутов, если пакет  
будет исполняемая программа?,  
Executable,  
EPakegs,  
Library,  
WProgram,  
Как указать зависимости для пакета? ,  
перечислить все пакеты, которые использованы в библиотеке,  
перечислить пакеты перед описанием каждого пакета,  
действие выполняется автоматически,  
нет верного ответа,  
Пакеты устанавливаются с помощью команды:,  
install,  
instal,  
instl,  
installing,  
Выберите команду для удаления библиотеки:,  
ghc-pkg unregister имя библиотеки,  
delete имя библиотеки,  
unregister имя библиотеки,  
uninstall имя библиотеки,  
Description - это: ,  
подробное описание назначения пакета.,  
статус версии библиотеки ,  
в этом поле можно через пробел указать дополнительные файлы,  
включаемые в пакет,  
нет верного ответа,  
Разработка игровой программы начинается с:,  
описания условий и наброска решения,  
описания условий,  
наброска решения,  
подключения модулей,  
С чего в Haskell начинают строить программы?,  
с каркаса - с типов и классов,  
с создания библиотек,

с создания пакетов,  
 нет верного ответа,  
 Какой подход был использован для написания игры пятнашки на Haskell?,  
 <сверху вниз>,,  
 <сверху вправо>,,  
 <справа налево>,,  
 <снизу вверх>,,  
 За что отвечает функция move в игре "Пятнашки" на языке Haskell?,  
 обновляет позицию после хода,  
 возвращает игру в начало,  
 приглашает игрока сделать ход,  
 функция принимает текущее положение,  
 В лямбда-исчислении программный код называется?,  
 Терм,  
 Функция,  
 Переменная,  
 Все перечисленное,  
 Лямбда-функции прежде всего предназначены для?,  
 описания анонимных функций и случаев, аналогичных частичному  
 применению (функций),  
 вычислений значений функций,  
 нет ответа,  
 -,  
 Результат выполнения кода  $\backslash x \rightarrow x + 1$  ?,  
 прибавляет к аргументу единицу,  
 изменяет аргумент,  
 вычитает из аргумента единицу,  
 нет правильного ответа,  
 Как должна выглядеть лямбда-функция для такой функции  $\text{noName } x = x + 1$  ?,  
 $\backslash x \rightarrow x + 1$ ,  
 $\backslash x \rightarrow x$ ,  
 $x = x + 1$ ,  
 $\backslash x = x + 1$ ,  
 какой тип вернет функция  $f :: [\text{Int}] \rightarrow [\text{Int}]$   $f = \text{filter } (\backslash x \rightarrow x > 2 \ \&\& \ x < 10 \ \&\& \ \text{even } x)$  ?,  
 [Int],  
 Bool,  
 Int,  
 [],  
 Функции в лямбда-исчислении называют?,  
 Абстракции,  
 каррированием.,  
 Нет правильного ответа,

Композицией,  
Какой символ заменяет `\` в Haskell?,

`\`,  
`=`,  
`->`,  
`*`,

Любую функцию  $n$  аргументов можно представить, как функцию одного аргумента, возвращающую другую функцию  $n-1$  аргументов. И по науке это называется?,

каррированием.,  
Нет правильного ответа,

Композицией,  
Декомпозицией ,

`add = \x -> \y -> x + y` является ли лямбда-функцией?,

Да ,

Нет,

Необходимы дополнения,

-,

В основе идеи  $\lambda$ -функций лежит?,

$\lambda$ -исчисление,

теория игр,

математическая статистика,

булева алгебра,

Какая запись в  $\lambda$ -функций соответствует этой записи  $\lambda x.x*x$  ?,

`5 -> (x * x)`,

`f = x*x`,

`f x = x*x`,

`5->x+x`,

Какая встроенная функция реализует этот код `\x -> x+1`?,

`Inc`,

`Add`,

`Filter`,

`Delete`,

Какая встроенная функция реализует этот код `\x y -> x+y`?,

`Add`,

`Inc`,

`Head`,

`null`,

К какому стилю определения функций можно отнести `add = (+)`?,  
бесточечный (point-free). ,

Точечный,

Нет правильного ответа,

локализованный,

С помощью чего можно имитировать локальные переменные?,

$\lambda$ -функцией,

термом,  
add,  
->,  
Кем было изобретено лямбда-исчисления?,  
Черчем,  
Тьюринг,  
Хаскелл,  
Шершников,  
Грамматика в Haskell является : по отношению к пространству лямбда-абстракций, let-выражений и условных выражений?,  
неоднозначной ,  
однозначной,  
неизвестно,  
нет ответа,  
Лямбда-нотация позволяет определять?,  
функции с той же легкостью, что и другие математические объекты,  
объекты,  
выражения,  
все перечисленное,  
Что вернет выполнение этой функции  $fst = \lambda(a, _) \rightarrow a$ ?,  
Первую часть картежа,  
Картеж,  
Список,  
Число,  
Как еще называют лямбда-выражения?,  
анонимными функции,  
программы,  
термины,  
нет правильного ответа

### ***Задания к лабораторным работам***

Описание лабораторной работы:

Лабораторная работа студентом выполняется за компьютером. Студент составляет соответствующую программу по заданию преподавателя, выполняет тестирование для проверки правильности работы программы. При сдаче работы преподавателю студент демонстрирует работу программы и отвечает на поясняющие вопросы преподавателя.

Варианты лабораторной работы:

Ниже приведены варианты заданий. Необходимо для заданного варианта написать исполняемую программу.

### **Лабораторная работа № 1.**

1. Заданы числа  $a$ ,  $b$  и  $c$ . Определить какое из чисел больше.
2. Напишите программу-модель анализа пожарного датчика в помещении, которая выводит сообщение «Пожарная ситуация», если температура (ее значение вводится с клавиатуры) в комнате превысила 60 градусов.
3. Первая бригада маляров за  $t_1$  час покрасила  $A$  квадратных метров стен, а вторая бригада за  $t_2$  часа покрасила  $B$  квадратных метров. У какой бригады производительность труда выше и на сколько?
4. Заданы числа  $a$  и  $b$ . Определить, эти числа одного или разных знаков.
5. Турист за день прошел  $A$  км. До обеда он шел  $t$  часов и прошел 20 км. Еще  $t_2$  часа он шел после обеда. Когда скорость туриста была выше: до обеда или после обеда?
6. Тело имеет массу  $M$  грамм и объем  $V$  см<sup>3</sup>. Будет ли оно плавать в жидкости, объем которой равен  $V_1$  см<sup>3</sup>, а масса  $M_1$  грамм? (Указание: найдите плотности тела и жидкости, а затем сравните их).
7. Валя и Вера собрались варить варенье из  $A$  кг смородины. По рецепту на 2 кг ягод нужно 3 кг сахара. Валя сказала, что им потребуется  $C$  кг сахара, а Вера - что  $P$  кг. Кто из них прав?
8. Масса 8 литров бензина 5,68 кг. Цистерна имеет объем 500 м<sup>3</sup>. Хватит ли ее, чтобы вместить  $A$  тонн бензина?
9. Кусок медного провода длиной 5 метров имеет массу 430 грамм. Чтобы провести проводку в квартире требуется  $C$  метров. Хватит ли для этой цели мотка провода массой  $M$  грамм?
10. Смекалкин, уходя в школу, вышел из дому на 3 минуты позже младшего брата. Расстояние до школы  $S$  метров. Смекалкин идет со скоростью  $V_1$  м/мин, а его брат –  $V_2$  м/мин. Догонит ли Смекалкин брата, прежде, чем тот придет в школу?
11. Составьте программу, проверяющую, верно ли утверждение, что введенное вами целое число является четным.
12. Составьте программу, проверяющую, верно ли утверждение, что введенное вами целое число делится без остатка на 3.
13. Составьте программу, проверяющую, верно ли утверждение, что сумма цифр введенного вами целого двузначного числа является четным.
14. Составьте программу, проверяющую, верно ли утверждение, что сумма цифр введенного вами целого трехзначного числа делится на 5.
15. Составьте программу, определяющую, пройдет ли график функции  $y = 5x^2 - 7x + 2$  через заданную точку с координатами  $(a, b)$ .
16. Даны три действительных числа  $a$ ,  $b$ ,  $c$ . Найти наибольшее из них.
17. Фермер намерен купить корову, дающую не менее  $L$  литров молока ежедневно с жирностью не менее  $K$  процентов. Написать алгоритм и программу выбора коровы.
18. Написать алгоритм и программу проверки, является ли данный четырехугольник параллелограммом.
19. Гнездо для ножки деревянной кровати имеет форму прямоугольной вырезки. Определить, подойдет ли данный четырехугольный брусок для этого гнезда.
20. Стороны одного прямоугольника равны  $A$  и  $B$ . Стороны другого равны  $X$  и  $Y$ . Написать алгоритм и программу проверки прямоугольников на равенство.

## Лабораторная работа № 2.



1. Составьте программу, выводящую на экран квадраты чисел от 10 до 20.
2. Составьте программу, которая вычисляет сумму чисел от 1 до 20.
3. Составьте программу, которая вычисляет сумму чисел от 1 до N. Значение N вводится с клавиатуры.
4. Составьте программу, которая вычисляет произведение чисел от 1 до N. Значение N вводится с клавиатуры.
5. С клавиатуры вводятся N чисел. Составьте программу, которая определяет количество отрицательных, количество положительных и количество нулей среди введенных чисел. Значение N вводится с клавиатуры.
6. Составьте программу, которая печатает таблицу перевода расстояний из дюймов в сантиметры (1 дюйм = 2,5 см) для значений длин от 1 до 20 дюймов.
7. Составить алгоритмы и программы перевода старинных русских мер длины, торгового и аптекарского веса (счетчик цикла меняется от 1 до 10):
  1. сажень в метры (1 сажень равна 2,1366 м)
  2. футов в метры (1 фут равен 0,3048 м)
  3. драхм в граммы (1 драхма равна 3,7325 г)
  4. унций в граммы (1 унция равна 29,86 г)
  5. фунтов в килограммы (1 фунт равен 0,40951 кг)
  6. аршинов в метры (1 аршин равен 0,7112 м)
  7. золотников в граммы (1 золотник равен 4,2657 г)
  8. дюймов в миллиметры (1 дюйм равен 25,3995 мм)
8. В сберкассу на трехпроцентный вклад положили S рублей. Какой станет сумма вклада через N лет?
9. Ввести с клавиатуры 10 пар чисел. Сравнить числа в каждой паре и напечатать большие из них.
10. Даны натуральные числа от 20 до 50. Напечатать те из них, которые делятся на 3, но не делятся на 5.
11. Даны натуральные числа от 35 до 87. Найти и напечатать те из них, которые при делении на 7 дают остаток 1, 2 или 5.
12. Даны натуральные числа от 1 до 50. Найти сумму тех из них, которые делятся на 5 или на 7.
13. Ввести с клавиатуры 10 чисел. Если среди них есть числа, большие 15, заменить их на 15. Напечатать все полученные числа.
14. Ввести с клавиатуры 10 чисел – положительных и отрицательных. Заменить все отрицательные числа их модулями и напечатать все полученные 10 чисел.
15. Напечатать те из двузначных чисел, которые делятся на 4, но не делятся на 6.
16. Найти произведение двузначных нечетных чисел, кратных 13.
17. Найти сумму чисел от 100 до 200, кратных 17.
18. Ввести с клавиатуры 10 чисел. Если квадрат числа меньше 100, напечатать число и его квадрат.
19. Составьте программу, которая вычисляет сумму квадратов чисел от 1 до введенного вами целого числа n.
20. Написать программу, которая по заданным значениям чисел a и b находит  $a^b$ . В запросе укажите допустимые значения этих переменных (например, если a – дробное, то b не может быть отрицательным).

### Лабораторная работа № 3.

1. Введите с клавиатуры в массив пять целочисленных значений. Выведите их в одну строку через запятую. Получите для массива среднее арифметическое.
2. Введите с клавиатуры пять целочисленных элементов массива X. Выведите на экран значения корней и квадратов каждого из элементов массива.
3. Создайте массив из пяти фамилий и выведите их на экран столбиком, начиная с последней.
4. Создайте массив из пяти фамилий и выведите их на экран те из них, которые начинаются с определенной буквы, которая вводится с клавиатуры.
5. Создайте массив A[1..7] с помощью генератора случайных чисел и выведите его на экран. Увеличьте все его элементы в 2 раза.
6. Создайте массив A[1..8] с помощью генератора случайных чисел с элементами от -10 до 10 и выведите его на экран. Подсчитайте количество отрицательных элементов массива.
7. Создайте массив A[1..12] с помощью генератора случайных чисел с элементами от -20 до 10 и выведите его на экран. Замените все отрицательные элементы массива числом 0.
8. Создайте целочисленный массив A[1..15] с помощью генератора случайных чисел с элементами от -15 до 30 и выведите его на экран. Определите самый большой элемент массива и его индекс.
9. С 8 до 20 часов температура воздуха измерялась ежедневно. Известно, что в течение этого времени температура понижалась. Определите, в котором часу была впервые отмечена отрицательная температура.
10. Данные о температуре воздуха за декаду ноября хранятся в массиве. Определить, сколько раз температура опускалась ниже -10 градусов.
11. Данные о температуре воды на Черноморском побережье за декаду сентября хранятся в массиве. Определить, сколько за это время было дней, пригодных для купания.
12. Данные о температуре воздуха и количестве осадков за декаду апреля хранятся в массивах. Определить количество осадков, выпавших в виде дождя и в виде снега за эту декаду.
13. Данные о температуре воздуха за декаду декабря хранятся в массиве. Определить, сколько раз температура была выше средней за эту декаду..
14. Данные о направлении ветра (северный, южный, восточный, западный) и силе ветра за декаду ноября хранятся в массиве. Определить, сколько дней дул южный ветер с силой, превышающей 8 м/с.
15. Создайте массив из 15 целочисленных элементов и определите среди них минимальное значение.

#### Лабораторная работа № 4.

1. На плоскости заданы n точек своими координатами  $(x_1, y_1)$ ,  $(x_2, y_2)$ . Составить программу вычисления максимального внутреннего и

минимального внешнего радиусов кольца с центром в начале координат, содержащего все точки.

2. Заданы координаты вершин двух треугольников. Определить пересекаются ли эти треугольники.

3. Создать список. Выполнить основные операции над списками (вывод первого элемента, вывод последнего элемента, вывод длины списка, вывод списка в обратном порядке).

4. Удалить заданный элемент из списка.

5. Вывести список в обратном порядке.

6. Создать список. Выполнить вывод первого элемента.

7. Создать список. Выполнить вывод последнего элемента.

8. Создать список. Выполнить вывод длины списка.

9. Создать список. Выполнить вывод списка в обратном порядке.

10. Написать программу вычисления факториала числа.

11. Написать программу вычисления суммы квадратов двух чисел.

12. Написать программу умножения числа на два, если число меньше 100.

13. Составить программу, которая вводит натуральное число  $N$  и выдает все трехзначные числа, сумма цифр которых равна  $N$ .

14. Задать два списка и получить произведения всех возможных комбинаций из элементов этих списков.

15. Найти длины сторон прямоугольного треугольника, удовлетворяющие следующим условиям: - длина каждой стороны меньше либо равна 10; - периметр треугольника 24.

### **Задания для контрольной работы**

Варианты контрольной работы. Группа делится на 3 варианта.

#### **Контрольная работа № 1**

Определите следующие функции:

1) Функция  $\max3$ , по трем целым возвращающая наибольшее из них.

2) Функция  $\min3$ , по трем целым возвращающая наименьшее из них.

3) Функция  $\text{sort2}$ , по двум целым возвращающая пару, в которой наименьшее из них стоит на первом месте, а наибольшее — на втором.

#### **Контрольная работа № 2**

Варианты контрольной работы. Группа делится на 3 варианта.

1) Функция  $\text{solve2}::\text{Double}\rightarrow\text{Double}\rightarrow(\text{Bool},\text{Double})$ , которая по двум числам, представляющим собой коэффициенты линейного уравнения  $ax + b = 0$ , возвращает пару, первый элемент которой равен  $\text{True}$ , если решение существует и  $\text{False}$  в противном случае; при этом второй элемент равен либо значению корня, либо 0.0.

2) Функция  $\text{isParallel}$ , возвращающая  $\text{True}$ , если два отрезка, концы которых задаются в аргументах функции, параллельны (или лежат на одной

прямой). Например, значение выражения `isParallel (1,1) (2,2) (2,0) (4,2)` должно быть равно `True`, поскольку отрезки  $(1, 1) - (2, 2)$  и  $(2, 0) - (4, 2)$  параллельны.

3) Функция `isIncluded`, аргументами которой служат параметры двух окружностей на плоскости (координаты центров и радиусы); функция возвращает `True`, если вторая окружность целиком содержится внутри первой.

### Перечень вопросов к зачету

1. История функционального программирования
2. Основные характеристики языка Haskell
3. Структура программы на языке Haskell.
4. Отладка программ на языке Haskell.
5. Типы данных.
6. Арифметические последовательности.
7. Функции для работы с целыми числами
8. Функции для работы с вещественными числами.
9. Функции для работы с символами.
10. Логические функции
11. Описание функции.
12. Образцы и сопоставление с образцом
13. Реализация ветвления.
14. Охраняющие выражения.
15. Локальные определения.
16. Аккумуляторы. Функция `error`
17. Строки. Замыкания списков
18. Базовые функции для работы со списками
19. Функции с условиями на список.
20. Функции, выполняющие некоторые действия над списком или его элементами
21. Кортежи. Классы типов данных
22. Графические средства языка Haskell
23. Язык Haskell и визуальное программирование
24. Лямбда функции
25. Взаимодействие Haskell с платформой .NET.

**3. Методические материалы, определяющие процедуры оценивания результатов обучения по дисциплине (модулю), описание шкал оценивания**

**Рейтинг-план дисциплины**

Виды учебной деятельности студентов	Балл за конкретное задание	Число заданий за семестр	Баллы	
			Минимальный	Максимальный
<b>Модуль 1</b>				
<b>Текущий контроль</b>			<b>0</b>	<b>25</b>
1. Устный опрос	5	1	0	5
2. Лабораторная работа	10	2	0	20
<b>Рубежный контроль</b>			<b>0</b>	<b>25</b>
1. Письменная контрольная работа	13	1	0	13
2 Тестовый контроль.	12	1	0	12
<b>Модуль 2</b>				
<b>Текущий контроль</b>			<b>0</b>	<b>25</b>
1. Устный опрос	5	1	0	5
2. Лабораторная работа	10	2	0	20
<b>Рубежный контроль</b>			<b>0</b>	<b>25</b>
1. Письменная контрольная работа	13	1	0	13
2 Тестовый контроль.	12	1	0	12
<b>Поощрительные баллы</b>			<b>0</b>	<b>10</b>
1. Студенческая олимпиада			0	5
2. Публикация статей			0	5
<b>Посещаемость (баллы вычитаются из общей суммы набранных баллов)</b>				
1. Посещение лекционных занятий			<b>0</b>	<b>-6</b>
2. Посещение практических (семинарских, лабораторных занятий)			<b>0</b>	<b>-10</b>
<b>Итоговый контроль</b>				
Зачет				

Результаты обучения по дисциплине (модулю) у обучающихся оцениваются по итогам текущего контроля количественной оценкой, выраженной в рейтинговых баллах. Оценке подлежит каждое контрольное мероприятие.

При оценивании сформированности компетенций применяется четырехуровневая шкала «неудовлетворительно», «удовлетворительно», «хорошо», «отлично».

Максимальный балл по каждому виду оценочного средства определяется в рейтинг-плане и выражает полное (100%) освоение компетенции.

Уровень сформированности компетенции «хорошо» устанавливается в случае, когда объем выполненных заданий соответствующего оценочного средства составляет 80-

100%; «удовлетворительно» – выполнено 40-80%; «неудовлетворительно» – выполнено 0-40%

Рейтинговый балл за выполнение части или полного объема заданий соответствующего оценочного средства выставляется по формуле:

Рейтинговый балл =  $k \times$  Максимальный балл,

где  $k = 0,2$  при уровне освоения «неудовлетворительно»,  $k = 0,4$  при уровне освоения «удовлетворительно»,  $k = 0,8$  при уровне освоения «хорошо» и  $k = 1$  при уровне освоения «отлично».

Оценка на этапе промежуточной аттестации выставляется согласно Положению о модульно-рейтинговой системе обучения и оценки успеваемости студентов УУНиТ:

На зачете выставляется оценка:

- зачтено - при накоплении от 60 до 110 рейтинговых баллов (включая 10 поощрительных баллов),
- не зачтено - при накоплении от 0 до 59 рейтинговых баллов.

При получении на экзамене оценок «отлично», «хорошо», «удовлетворительно», на зачёте оценки «зачтено» считается, что результаты обучения по дисциплине (модулю) достигнуты и компетенции на этапе изучения дисциплины (модуля) сформированы.