

Дата: 2020/11/23
Предмет: Информатика
Тема: Элементы программирования
Тип занятия: Лабораторная работа
Группа: ТК-20

Порядок выполнения работы: решить задачи в среде программирования Thonny, сохранить в архив, выложить в личный кабинет.

Критерии оценки выполнения задания: работа считается зачѐнной, если задания выполнены правильно.

Порядок распределения заданий между студентами:
студенты выполняют задания согласно распределению вариантов.

- 1 Батомункуев Баир Тумэнович
- 2 Богомолов Федор Олегович
- 3 Бронникова Кристина Романовна
- 4 Виткаускас Павел Анатольевич
- 5 Власов Даниил Евгеньевич
- 6 Гурулева Дарья Викторовна
- 7 Дамдинжапов Тимур Владимирович
- 8 Дашидондоков Алдар Борисович
- 9 Доржиева Галина Шимитовна
- 10 Зверев Павел Васильевич
- 1 Иринчинова Валентина Баясхалановна
- 2 Концов Никита Витальевич
- 3 Курицын Сергей Викторович
- 4 Марковцев Константин Игоревич
- 5 Позднякова Александра Витальевна
- 6 Попов Александр Александрович
- 7 Попов Михаил Владимирович
- 8 Потапов Игорь Андреевич
- 9 Рыбаков Алексей Сергеевич
- 10 Сидаш Алина Андреевна
- 1 Стукалов Дмитрий Игоревич
- 2 Стуков Антон Валерьевич
- 3 Тарханов Денис Алексеевич
- 4 Тарханова Любовь Сергеевна
- 5 Толстоногов Виктор Викторович

- 6 Толстоногова Любовь Сергеевна
- 7 Тунгаланов Базар Очирович
- 8 Цыпылов Иван Владимирович
- 9 Цэдашиева Жаргалма Амгалановна
- 10 Щербатюк Илья Максимович
- 1 Яковлев Николай Викторович

Теоретический материал

1. Вступление

Представьте, что перед вами поставлена задача: даны анкеты учащихся некоторого класса и требуется подсчитать количество мальчиков и девочек в этом классе. Можно предложить следующий способ решения. На школьной доске отведём место для двух чисел – количества мальчиков и количества девочек, и для начала в этих местах запишем по нулю. Далее будем перебирать анкету за анкетой и смотреть, какой пол указан в просматриваемой анкете. Если пол мужской, то будем заменять число на доске, показывающее количество мальчиков, числом, большим на единицу. Если же пол женский, то такую операцию будем выполнять с числом на доске, показывающим количество девочек. Как только мы переберём все анкеты, на доске останутся два числа, равные количеству мальчиков и девочек в классе.

В данном примере отражены следующие особенности программирования.

- Чтобы решить задачу, необходимо придумать метод, который за конечное количество действий приводит к решению (либо к осознанию, что решения нет). Такой метод называется *алгоритмом*.
- При решении задачи приходится работать с данными, которые имеют свой *тип*. К примеру, данными являются переменные *числа*, записанные на доске, а также *упорядоченное множество* анкет, содержащих *текстовую* информацию о том, является ли ученик мальчиком или девочкой.
- В ходе работы алгоритма иногда приходится повторить одно и то же действие несколько раз. Это называется *циклом*. В нашем примере в цикле просматриваются анкеты учеников и меняются числа на доске.
- В некоторых случаях выполняемые действия зависят от тех или иных условий. Такая алгоритмическая конструкция называется *ветвлением*. В примере, приведённом выше, изменение первого или второго числа на доске зависело от записи в просматриваемой анкете.
- Наконец заметим, что алгоритм, приведённый в примере, осуществлял *исполнитель* – человек. Поэтому очень важно, на каком языке записан алго-

ритм. Компьютеры «понимают» только язык машинных кодов. Но с машинными кодами человеку очень трудно иметь дело. Поэтому были созданы *языки программирования*, которые позволяют по особым правилам с помощью словесных конструкций записать алгоритм и сохранить так называемый *исходный код* программы в обычном текстовом файле. Далее этот файл обрабатывается или *транслируется* с помощью специальной программы, которая либо создаёт файл машинных кодов, который уже выполняется в операционной системе (в этом случае метод обработки исходного кода называется *компиляцией*), либо сама выполняет записанные в текстовом файле команды друг за дружкой и таким образом возвращает некоторый результат (в данном случае метод обработки исходного кода называется *интерпретацией*).

Надеемся, что после данного рассуждения становятся понятны слова Никлауса Вирта, создателя языка программирования pascal:

алгоритмы + структуры данных = программы.

Далее мы рассмотрим язык программирования python. Программы, записанные на этом языке, являются *интерпретируемыми*.

Для написания исходных текстов программ нужно скачать и установить бесплатные интегрированные среды программирования:

- либо официальный релиз Python и IDLE,
- либо Thonny,
- либо Python + Jupyter Notebook.

Наиболее простая IDE – Thonny. Её можно скачать по ссылке:

<https://github.com/thonny/thonny/releases/download/v3.2.7/thonny-3.2.7.exe>

2. Структуры данных и ввод-вывод

2.1. Хранение данных в памяти во время работы программы

Данные, используемые в программе, должны находиться в памяти компьютера. Чтобы иметь доступ к области памяти с данными, эта область должна быть поименована (с адресом памяти должно быть связано некоторое имя).

В ходе работы программы содержимое поименованной области памяти может изменяться. В этом случае данная область памяти и имя, с ней связанное, называется *переменной*. Если же содержимое области памяти не меняется, то в этом случае мы имеем дело с *постоянной величиной* или *константой*.

Выделяемая область имеет свой размер (в байтах). К примеру, мы можем выделить область величиной один байт и поместить туда двоичное значение 00100001.

Но что оно будет означать? Это может быть, к примеру, десятичное число 33, либо символ !. Программа должна чётко понимать, с данными какого *типа* она работает.

Имеется два способа задания типа данных: *статический*, когда тип данных указывается при задании переменной или константы, и *динамический*, когда тип данных сопоставляется с константой или переменной при помещении в неё каких-либо конкретных данных. Языки pascal и c++ являются языками со статической типизацией. Язык python – это язык с динамической типизацией.

3. Основы языка Python

3.1. Встроенные типы данных. Простейший ввод-вывод

3.1.1. Встроенные типы данных

В программном коде мы будем использовать следующие встроенные классы (типы) данных:

Тип данных	Пояснение
<code>int</code>	целое число (с произвольным количеством знаков)
<code>float</code>	число с плавающей точкой
<code>str</code>	строка
<code>bool</code>	логическая величина, принимающая значения <code>True</code> (истина) или <code>False</code> (ложь)
<code>tuple</code>	кортеж – упорядоченный (перенумерованный) набор, состоящий из разнотипных элементов
<code>range</code>	последовательность чисел, обычно используемая в циклах
<code>list</code>	список – упорядоченный (перенумерованный) набор разнотипных элементов
<code>set</code>	множество – неупорядоченный (не перенумерованный) набор разнотипных элементов
<code>dict</code>	словарь – набор разнотипных элементов, перенумерованных не числами, а уникальными значениями – ключами

Первые шесть типов данных являются *неизменяемыми*. Смысл неизменяемости можно пояснить следующим образом. Сравним работу с переменными в Pascal и Python. Когда переменная определяется в языке Pascal (в разделе определения переменных `var`), то ей выделяется область памяти и с этой областью памяти ассоциируется имя переменной. Содержимое этой области памяти может изменяться, но имя, ассоциированное с областью памяти, сохраняется.

Когда переменная определяется в языке Python (в момент первого появления в коде в присваивании или вводе с клавиатуры), то ей выделяется область памяти и с этой областью памяти ассоциируется имя переменной. Если мы хотим поменять значение неизменяемой переменной, то новому значению выделится другая область памяти, имя переменной будет перенесено на новую область памяти, а старое значение будет потеряно.

Списки, кортежи, множества и словари являются изменяемыми в том плане, что мы имеем возможность менять их содержимое, но при этом адрес переменной указанных типов не меняется (но меняются адреса их элементов).

3.1.2. Замечание по поводу оформления программного кода

Программный код, приводимый в издании, будет оформляться моноширинным шрифтом, возможно, меньшего размера, чем основной текст, и располагаться в ограничивающей рамке. Знаки операторов будут обрамляться пробелами. Ниже приведён пример кода программы и выводимого ею результата. Однострочные комментарии в программе пишутся после знака #.

Листинг 1. Пример программного кода

```
# суммирование матриц a и b
import numpy
a = numpy.array([
    [2, 5, -4],
    [3, 7, 2]
])
b = numpy.array([
    [ 0, 3, 1],
    [-5, 7, -4]
])
print(a+b)
```

Результат:

```
[[ 2  8 -3]
 [-2 14 -2]]
```

3.1.3. Способы задания переменных

Переменная может быть задана с помощью оператора присваивания. Ниже приведён пример задания целочисленной переменной *x*, вещественной переменной *y*, строковых переменных *z* и *w*, списка *a*, кортежа *k*, булевой переменной *b*, множества *s*, словаря *d*.

В Python поддерживается множественное присваивание. В примере ниже оно выполнено в последней строке.

Листинг 2. Задание переменных с помощью присваивания

```
x = 2
y = 2.1545
z = 'строка'
w = "строка"
a = [1, "d", 2.36]
k = (1, 3.14, "d")
b = True
s = {1, 2, "3"}
d = {"Иван": 28, "Пётр": 45}
alpha, beta = 30, 60
```

Переменные можно ввести с клавиатуры. Это делается с помощью функции `input`. Необязательным аргументом этой функции является строка – приглашение пользователю. По умолчанию данная функция возвращает строковую величину. Следовательно, чтобы ввести с клавиатуры целое или вещественное число, нужно результат функции `input` преобразовать с помощью функций `int` и `float` в целый и вещественный тип, соответственно. Ниже показан ввод строковой переменной `s`, целочисленной переменной `n` и вещественной переменной `f`

Листинг 3. Ввод переменных с клавиатуры

```
s = input("Введите строку s: ")
n = int(input("Введите целое число n: "))
f = float(input("Введите вещественное число f: "))
```

Булеву переменную более корректно задавать присваиванием. Заметим, что конструкция `b = bool(input())` будет работать. При этом, если с клавиатуры введена непустая строка символов, то в `b` будет записано `True`. Если же была введена пустая строка, то в `b` будет записано `False`.

3.1.4. Преобразование типов данных

Как уже было показано в предыдущем подпункте, строковые данные, вводимые с клавиатуры, можно преобразовать в целочисленные и вещественные (если введённые строки можно в такие данные преобразовать). При этом применялись функции, совпадающие по имени с именами соответствующих типов данных. Такой принцип действует для всех указанных выше типов. К примеру, можно строку преобразовать в список, а список – во множество. Множество содержит только неповторяющиеся значения, и это можно использовать, например, при подсчёте уникальных элементов списка или строки.

Листинг 4. Преобразование типов

```
myString = "информатика"
myList = list(myString)
print(myList)
mySet = set(myList)
print(mySet)
```

Результат:

```
['и', 'н', 'ф', 'о', 'р', 'м', 'а', 'т', 'и', 'к', 'а']
{'к', 'р', 'н', 'ф', 'м', 'а', 'о', 'т', 'и'}
```

3.1.5. Допустимые имена переменных

В Python имена переменных должны подчиняться правилу: имя может состоять из букв латиницы, цифр и знака подчёркивания, причём, цифра не может быть первым символом в имени.

Желательно, чтобы имя переменной соответствовало её смысловой нагрузке в программе. Так, `value` более приемлемо, чем `v`. Вместе с тем имя не должно быть слишком длинным.

Прописные и строчные буквы в именах различаются. Так, имена `namedVertices` и `namedvertices` являются различными.

Следует избегать совпадений с уже имеющимися именами, определёнными в языке. Например, не стоит давать переменной, хранящей максимальное из двух чисел, имя `max`, так как есть функция с таким же именем.

3.1.6. Вывод текста на экран

Для вывода текстовой информации служит функция `print`, аргументами которой могут быть величины различных типов. Количество аргументов может быть произвольным.

При решении вычислительных задач будет необходимо уметь выводить вещественные значения с определённым количеством знаков после запятой. Для этого можно применять метод строки `format`.

В примере ниже показано, как во второй строке число `x` выводится в таком же виде, как оно было задано; в третьей строке число выводится в 10 позициях, из которых 3 позиции отведены на дробную часть; в четвёртой строке количество позиций на экране подбирается автоматически, из них 2 позиции отведены на дробную часть; в пятой строке число выводится в экспоненциальной форме (то есть, в виде $1.235e+02 = 1.235 \cdot 10^2$). Заметим, что целая и дробные части вещественного числа разделяются точкой.

Листинг 5. Форматный вывод значений

```
x = 123.45678
print("Значение: ",x)
print("Значение: {:.10.3f}".format(x))
print("Значение: {:.0.2f}".format(x))
print("Значение: {:.10.3e}".format(x))
```

Результат:

```
Значение: 123.45678
Значение: 123.457
Значение: 123.46
Значение: 1.235e+02
```

Иной способ вывода на экран числовых значений и результатов вычисления формул состоит в использовании *f*-строк. Рассмотрим пример. Пользователь вводит с клавиатуры некоторое вещественное число, а программа выводит квадрат этого числа, оставляя 2 знака после десятичной точки.

Листинг 6. Пример использования *f*-строк

```
n = float(input("Введите число: "))
print(f"Квадрат числа: {n**2 :0.2f}")
```

Результат:

```
Введите число: 3.44545
Квадрат числа: 11.87
```

Видим, что перед строкой ставится префикс *f*, а внутри строки в фигурных скобках помещено выражение, которое вычисляет квадрат введенного числа и к результату вычисления этого выражения применён форматный вывод.

3.1.7. Три полезные функции

В Python имеются три полезные функции:

Функция	Пояснение
<code>id(obj)</code>	возвращает идентификатор объекта <code>obj</code> (адрес объекта в памяти)
<code>type(obj)</code>	возвращает тип объекта <code>obj</code>
<code>dir(obj)</code>	возвращает список атрибутов объекта <code>obj</code>

Листинг 7. Пример использования функций `type`, `dir`, `id`

```
x = [1, 2, 4]
print(id(x))
print(type(x))
print(dir(x))
```

Результат:

```
45460136
<class 'list'>
['__add__', '__class__', '__contains__', '__delattr__', '__delitem__',
 '__dir__', '__doc__', '__eq__', '__format__', '__ge__',
 '__getattr__', '__getitem__', '__gt__', '__hash__', '__iadd__',
 '__imul__', '__init__', '__init_subclass__', '__iter__', '__le__',
 '__len__', '__lt__', '__mul__', '__ne__', '__new__', '__reduce__',
 '__reduce_ex__', '__repr__', '__reversed__', '__rmul__',
 '__setattr__', '__setitem__', '__sizeof__', '__str__',
 '__subclasshook__', 'append', 'clear', 'copy', 'count', 'extend',
 'index', 'insert', 'pop', 'remove', 'reverse', 'sort']
```

Можно заметить, что функция `type` выводит не просто наименование типа данных, но именует его `class`. Суть в том, что в Python самым полнейшим образом реализован объектно-ориентированный подход, и, как говорят, в Python всё является классом. Целое число – это класс, вещественное число – класс, и пр. Даже функция – это тоже класс. Поэтому далее, говоря об элементах программ, мы будем упоминать слово объект или экземпляр класса. Так как у классов определяются атрибуты (свойства и методы), то в дальнейшем мы тоже будем использовать указанные термины.

Функцию `dir` часто используют для изучения свойств и методов объектов, а также для вывода списка имён, определённых в текущей области видимости или в некотором модуле (когда аргументом функции является имя модуля).

Функция `id` полезна для понимания того, как в Python реализуется оператор присваивания. Опять же можно провести сравнение языков Pascal и Python.

Если мы в Pascal присваиваем одной переменной, например, *x*, значение другой переменной, например, *y*, то из области памяти переменной *y* в область памяти переменной *x* переносится содержимое. Имена переменных остаются закреплёнными за выделенными областями памяти.

Если мы в Python присваиваем одной переменной, например, *x*, значение другой переменной, например, *y*, то с областью памяти, в которой находится значение, ассоциированное с именем *y*, будет и ассоциировано имя *x*, так что два имени переменных будут указывать на одну и ту же область памяти. Указанный подход к понятию переменной понуждает нас более осторожно относиться к применению операций присваивания.

Листинг 8. *Что происходит при присваивании*

```
x = [1,2,3]
y = ["a","b"]
print(x,y)
print(id(x),id(y))
x = y
print(x,y)
print(id(x),id(y))
```

Результат:

```
[1, 2, 3] ['a', 'b']
45525752 45369184
['a', 'b'] ['a', 'b']
45369184 45369184
```

Так как в Python имена переменных – это «ярлыки» к областям памяти, то можно одновременно «перевешивать» эти «ярлыки» для обмена значениями переменных без использования промежуточной переменной.

Листинг 9. *Обмен значениями переменных*

```
frst = "Петя"
scnd = "Вася"
print(frst, scnd)
frst, scnd = scnd, frst
print(frst, scnd)
```

Результат:

3.2. Арифметические операторы

Перечислим арифметические операторы, используемые в Python.

Оператор	Пояснение
+	сложение
-	вычитание
*	умножение
/	деление
**	возведение в степень
//	неполное частное целых чисел
%	остаток от деления одного целого числа на другое

Листинг 10. Арифметика в Python

```
x = 17
y = 7
print('x + y =', x + y)
print('x - y =', x - y)
print('x * y =', x * y)
print('x / y =', x / y)
print('x в степени y =', x ** y)
print('a div b =', x // y)
print('a mod b =', x % y)
```

Результат:

```
x + y = 24
x - y = 10
x * y = 119
x / y = 2.4285714285714284
x в степени y = 410338673
a div b = 2
a mod b = 3
```

3.3. Модули и пакеты

Язык Python позволяет собирать пользовательские функции в отдельные файлы с расширением `py` (впрочем, такое расширение имеют все файлы с исходным

кодом Python). Файлы, в которых собраны определения функций и/или классов, называются *модулями*.

Несколько модулей составляют *пакет*. Физически пакет – это папка, в которую сложены файлы модулей, а также файл `__init__.py`, в котором может храниться служебная информация (данный файл может быть и пустым, но сам файл с таким именем обязан быть в папке пакета). Именем пакета является имя папки пакета.

Чтобы использовать содержимое, определённое в пакете или модуле, используется инструкция `import`. В следующем пункте мы перечислим функции, определённые в модуле `math` и далее приведём пример создания собственного пакета и его использования.

3.3.1. Модуль `math`

Чтобы использовать математические функции, необходимо подгрузить модуль `math`. Перечислим функции, определённые в этом модуле.

Функция	Пояснение
<code>fabs(x)</code>	абсолютная величина (модуль) числа x
<code>floor(x)</code>	целая часть «снизу», наибольшее целое число, меньшее или равное x
<code>ceil(x)</code>	целая часть «сверху», наименьшее целое число, большее или равное x
<code>exp(x)</code>	экспонента e^x
<code>log(x)</code>	натуральный логарифм $\ln x$
<code>log(x, a)</code>	логарифм по произвольному основанию $\log_a x$
<code>pow(x, n)</code>	степень x^n
<code>sqrt(x)</code>	квадратный корень $\sqrt{x}$
<code>pi</code>	число $\pi \approx 3.14159$
<code>e</code>	число $e \approx 2.71828$
<code>sin(x)</code>	синус $\sin x$
<code>cos(x)</code>	косинус $\cos x$
<code>tan(x)</code>	тангенс $\operatorname{tg} x$
<code>asin(x)</code>	арксинус $\arcsin x$
<code>acos(x)</code>	арккосинус $\arccos x$
<code>atan(x)</code>	арктангенс $\operatorname{arctg} x$
<code>radians(x)</code>	перевод угла x из градусной меры в радианную
<code>degrees(x)</code>	перевод угла x из радианной меры в градусную

Как получить доступ к данным функциям? Если мы запишем инструкцию `import math`, то получим доступ ко всем функциям и объектам, определённым в модуле. Вызов функций в тексте программы осуществляется с помощью селектора (имени модуля, после которого идет точка, а за ней следует имя функции). В примере ниже показано вычисление значения выражения $\sin \frac{\pi}{4}$.

Листинг 11. Использование математического модуля

```
import math
print(math.sin(math.pi/4))
```

Результат:

0.7071067811865475

В следующем примере показано вычисление площади треугольника по формуле: $s = 2R^2 \sin \alpha \sin \beta \sin \gamma$ по заданному радиусу описанной окружности R и двум углам α и β , заданным в градусах (третий угол γ вычисляется по первым двум).

Листинг 12.

```
import math
# ввод данных
r = float(input("Введите радиус описанной окружности: "))
alpha = float(input("Введите первый угол треугольника (в градусах): "))
beta = float(input("Введите второй угол треугольника (в градусах): "))
# вычисление третьего угла
gamma = 180 - alpha - beta
# перевод из градусов в радианы
alpha = math.radians(alpha)
beta = math.radians(beta)
gamma = math.radians(gamma)
# вычисление площади и вывод её на экран
s = 2 * r * math.sin(alpha) * math.sin(beta) * math.sin(gamma)
print(f"Площадь треугольника: s")
```

Результат:

```
Введите радиус описанной окружности: 3
Введите первый угол треугольника (в градусах): 36
Введите второй угол треугольника (в градусах): 47
Площадь треугольника: 2.560048006432965
```

Инструкция `import` позволяет задать псевдоним загружаемому модулю (это делается, к примеру, если имя модуля слишком длинно). Тогда в тексте программы вместо исходного имени модуля нужно использовать псевдоним.

Листинг 13. Использование псевдонима

```
import math as m
print(m.sin(m.pi/4))
```

Чтобы получить доступ только к некоторой функции, определённой в модуле, можно использовать инструкцию

`from math import имя_функции`

В этом случае в тексте программы селектор не используется.

Листинг 14. Загрузка отдельных функций

```
from math import sin, pi
print(sin(pi/4))
```

Необходимо заметить, что селектор также не используется в случае, когда загрузка содержимого модуля была осуществлена с использованием инструкции `from math import *`

Листинг 15. Загрузка всех функций

```
from math import *
print(sin(pi/4))
```

Задания

Лабораторная работа №1. Вычисления

Вариант 1.

- 1.1. Вычислить: $c = \sqrt{a^2 + b^2}$. Показать результат с двумя знаками после запятой.
- 1.2. Вычислить: $c = \frac{a^2 + b^3}{a^4 + b^5}$. Показать результат с тремя знаками после запятой.
- 1.3. Вычислить: $z = \sin^2 x - \cos^2 y + \operatorname{tg} y - \operatorname{ctg} x$. Углы x и y заданы в градусах.
- 1.4. По заданным длинам a, b, c сторон треугольника вычислить длину медианы треугольника, проведённой к стороне a :

$$m = \sqrt{\frac{b^2 + c^2}{2} - \left(\frac{a}{2}\right)^2}$$

Вариант 2.

2.1. Вычислить: $c = \sqrt{a^2 - b^2}$. Показать результат с тремя знаками после запятой.

2.2. Вычислить: $c = \frac{a^3 - b^5}{a^4 - b^2}$. Показать результат с четырьмя знаками после запятой.

2.3. Вычислить: $z = \sin^3 x - \cos^3 y - \operatorname{tg} y \cdot \operatorname{ctg} x$. Углы x и y заданы в градусах.

2.4. По заданным длинам a, b, c сторон треугольника вычислить длины частей, на которые биссектриса, проведённая к стороне a , делит эту сторону:

$$x = \frac{ca}{c + b}, \quad y = \frac{ba}{c + b}$$

Вариант 3.

3.1. Вычислить: $c = (\sqrt{a} + \sqrt{b})^2$. Показать результат с четырьмя знаками после запятой.

3.2. Вычислить: $c = a^3 + \frac{b^4}{a^5 + b^2}$. Показать результат с одним знаком после запятой.

3.3. Вычислить: $z = \sin^2 x + \cos^3 y + \operatorname{tg} y : \operatorname{ctg} x$. Углы x и y заданы в градусах.

3.4. По заданным длинам b, c сторон треугольника и углу α (задан в градусах) вычислить длину биссектрисы треугольника, проведённой к стороне a :

$$l = \frac{2cb}{c + b} \cos \frac{\alpha}{2}$$

Вариант 4.

4.1. Вычислить: $c = (\sqrt{a} - \sqrt{b})^2$. Показать результат с одним знаком после запятой.

4.2. Вычислить: $c = a^3 - \frac{b^5}{a^4 - b^2}$. Показать результат с двумя знаками после запятой.

4.3. Вычислить: $z = \sin^3 x + \cos^2 y - \operatorname{tg} y : \operatorname{ctg} x$. Углы x и y заданы в градусах.

4.4. По заданным длинам a, b, c сторон треугольника вычислить длину биссектрисы треугольника, проведённой к стороне a :

$$l = \sqrt{bc \left(1 - \frac{a^2}{(b + c)^2} \right)}$$

Вариант 5.

5.1. Вычислить: $c = a^2 + 2\sqrt{a} \cdot \sqrt{b} + b^2$. Показать результат с двумя знаками после

запятой.

5.2. Вычислить: $c = \frac{a^3 + b^2}{a^5} + b^4$. Показать результат с тремя знаками после запятой.

5.3. Вычислить: $z = \sin x \cdot \cos y + \operatorname{tg}^2 y - \operatorname{ctg}^3 x$. Углы x и y заданы в градусах.

5.4. По заданным длинам a , b сторон треугольника и углу γ (задан в градусах) вычислить длину стороны c треугольника:

$$c = \sqrt{a^2 + b^2 - 2ab \cos \gamma}$$

Вариант 6.

6.1. Вычислить: $c = a^2 - 2\sqrt{a} \cdot \sqrt{b} + b^2$. Показать результат с тремя знаками после запятой.

6.2. Вычислить: $c = \frac{a^3 - b^2}{a^4} - b^5$. Показать результат с четырьмя знаками после запятой.

6.3. Вычислить: $z = \sin x \cdot \cos y - \operatorname{tg}^3 y + \operatorname{ctg}^2 x$. Углы x и y заданы в градусах.

6.4. По заданным длинам a , b , c сторон треугольника вычислить его полупериметр $p = \frac{a + b + c}{2}$ и площадь:

$$s = \sqrt{p(p - a)(p - b)(p - c)}$$

Вариант 7.

7.1. Вычислить: $c = \sqrt{a^2 + 2ab + b^2}$. Показать результат с четырьмя знаками после запятой.

7.2. Вычислить: $c = \frac{a^2 - b^3}{a^4 + b^5}$. Показать результат с тремя знаками после запятой.

7.3. Вычислить: $z = \sin x : \cos y - \operatorname{tg}^3 y + \operatorname{ctg}^3 x$. Углы x и y заданы в градусах.

7.4. По заданным углам треугольника α , β (заданы в градусах) и радиусу описанной окружности r вычислить третий угол γ треугольника и площадь треугольника:

$$s = 2 \cdot r^2 \cdot \sin \alpha \cdot \sin \beta \cdot \sin \gamma$$

Вариант 8.

8.1. Вычислить: $c = \sqrt{a^2 - 2ab + b^2}$. Показать результат с одним знаком после запятой.

8.2. Вычислить: $c = \frac{a^2 + b^5}{a^4 - b^3}$. Показать результат с двумя знаками после запятой.

8.3. Вычислить: $z = \sin x : \cos y + \operatorname{tg}^2 y + \operatorname{ctg}^2 x$. Углы x и y заданы в градусах.

8.4. По заданным длинам b, c сторон треугольника и углу α (задан в градусах) вычислить площадь треугольника:

$$s = \frac{1}{2}bc \sin \alpha$$

Вариант 9.

9.1. Вычислить: $c = \sqrt{\frac{1}{a^2} + \frac{1}{b^2}}$. Показать результат с тремя знаками после запятой.

9.2. Вычислить: $c = \frac{a^2 + b^3}{a^4} - b^5$. Показать результат с четырьмя знаками после запятой.

9.3. Вычислить: $z = \sin x \cdot \cos y - \operatorname{tg}^2 y + \operatorname{ctg}^2 x$. Углы x и y заданы в градусах.

9.4. По заданным длинам a, b сторон параллелограмма и углу γ между ними (задан в градусах) вычислить площадь параллелограмма:

$$s = bc \sin \alpha$$

Вариант 10.

10.1. Вычислить: $c = \sqrt{\frac{1}{a^2} - \frac{1}{b^2}}$. Показать результат с двумя знаками после запятой.

10.2. Вычислить: $c = \frac{a^3 - b^2}{a^4} + b^5$. Показать результат с одним знаком после запятой.

10.3. Вычислить: $z = \sin x \cdot \cos y - \operatorname{tg}^3 y - \operatorname{ctg}^3 x$. Углы x и y заданы в градусах.

10.4. По заданным длинам u, v диагоналей выпуклого четырёхугольника и углу δ между ними (задан в градусах) вычислить площадь четырёхугольника:

$$s = \frac{1}{2}uv \sin \delta$$