Программирование на языке Python. Уровень 1.

Исключения и обработка ошибок.

Обработка исключений. Типы ошибок.

- **Исключения** — это извещения интерпретатора, возбуждаемые в случае возникновения ошибки в программном коде или при наступлении какого-либо события. Если в коде не предусмотрена обработка исключения, выполнение программы прерывается, и выводится сообщение об ошибке.
- В программе могут встретиться три типа ошибок:

♦ ***синтаксические*** — это ошибки в имени оператора или функции, отсутствие закрывающей или открывающей кавычек и т. д., то есть ошибки в синтаксисе языка. Как правило, интерпретатор предупредит о наличии такой ошибки, а программа не будет выполняться совсем.

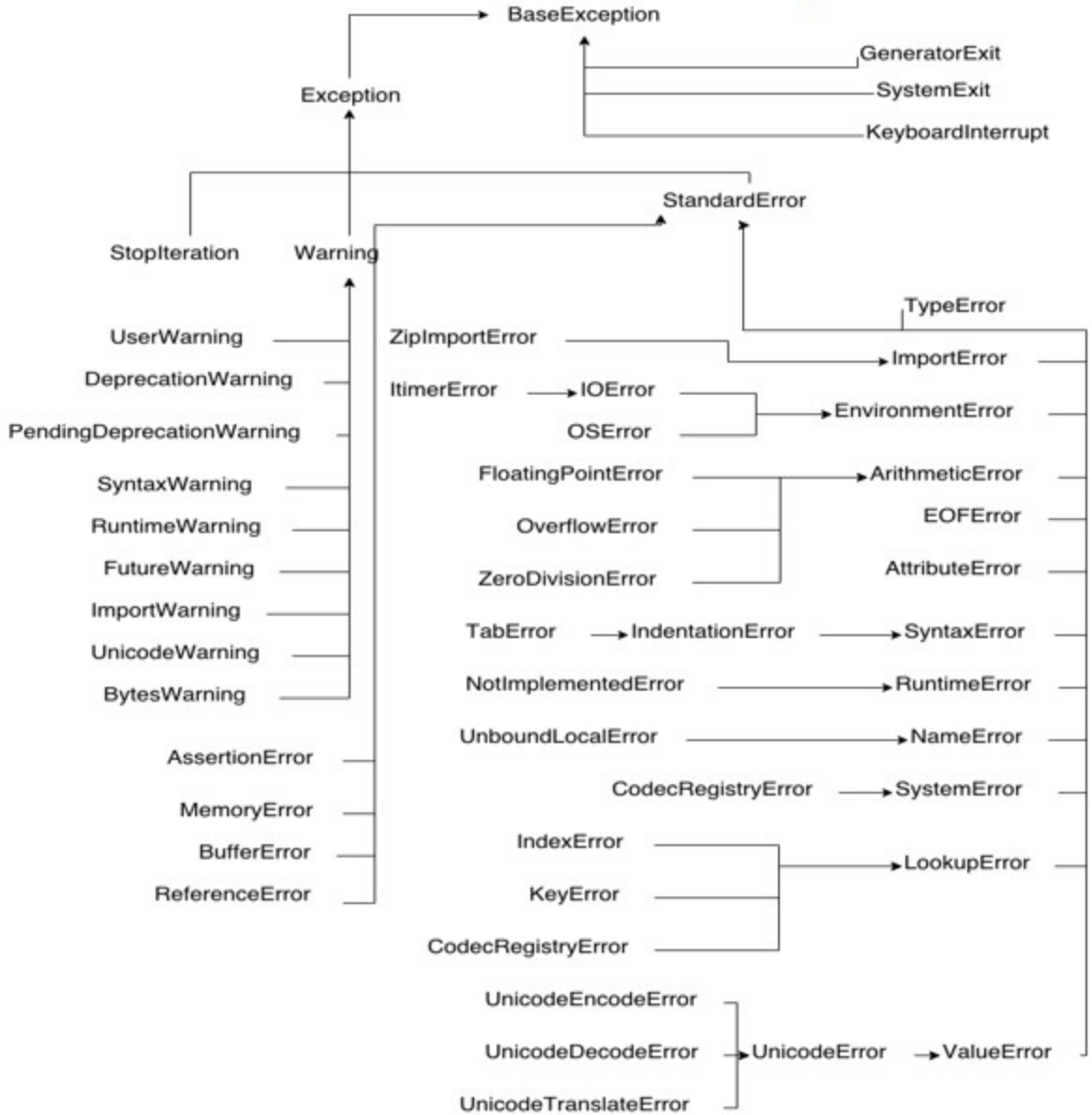
♦ ***логические*** — это ошибки в логике программы, которые можно выявить только по результатам ее работы. Как правило, интерпретатор не предупреждает о наличии такой ошибки, и программа будет успешно выполняться, но результат ее выполнения окажется не тем, который был ожидаем.

Выявить и исправить логические ошибки весьма трудно!

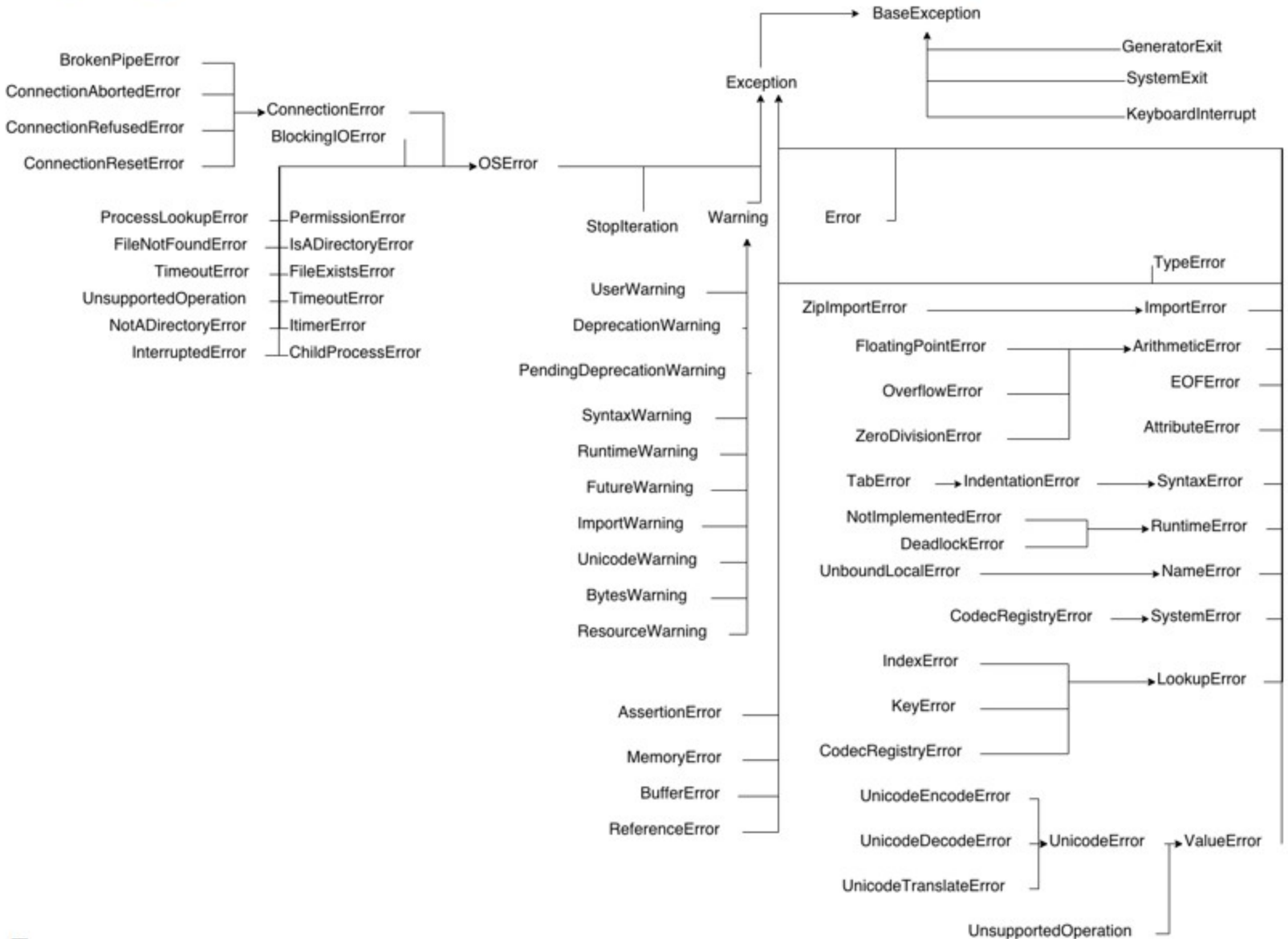
♦ ***ошибки времени выполнения*** — это ошибки, которые возникают во время работы программы. Причиной являются события, не предусмотренные программистом.

Концепция исключений

Иерархия исключений в Python 2.x



Иерархия исключений в Python 3.x



Основные классы встроенных исключений

- BaseException — является классом самого верхнего уровня и базовым для всех прочих классов исключений;
- Exception — базовый класс для большинства встроенных в Python исключений;
- AttributeError — попытка обращения к несуществующему атрибуту объекта;
- EOFError — возбуждается функцией `input()` при достижении конца файла;
- ImportError — невозможно импортировать модуль или пакет;
- IndentationError — неправильно расставлены отступы в программе;
- IndexError — указанный индекс не существует в последовательности;
- KeyError — указанный ключ не существует в словаре;
- KeyboardInterrupt — нажата комбинация клавиш <Ctrl>+<C>;

Основные классы встроенных исключений

- MemoryError — интерпретатору не хватает оперативной памяти;
- NameError — попытка обращения к идентификатору до его определения;
- OSError — базовый класс для всех исключений, возбуждаемых в ответ на возникновение ошибок в операционной системе (отсутствие запрошенного файла, недостаток места на диске и прочие системные исключения);
- OverflowError — число, получившееся в результате выполнения арифметической операции, слишком велико, чтобы Python смог его обработать;
- RecursionError — превышено максимальное количество проходов рекурсии;
- RuntimeError — неклассифицированная ошибка времени выполнения;
- StopIteration — возбуждается методом `__next__()` как сигнал об окончании итераций;
- SyntaxError — синтаксическая ошибка;

Основные классы встроенных исключений

- SystemError — ошибка в самой программе интерпретатора Python;
- TabError — в исходном коде программы встретился символ табуляции, использование которого для создания отступов недопустимо;
- TypeError — тип объекта не соответствует ожидаемому;
- UnboundLocalError — внутри функции переменной присваивается значение после обращения к одноименной глобальной переменной;
- UnicodeDecodeError — ошибка преобразования последова-тельности байтов в строку;
- UnicodeEncodeError — ошибка преобразования строки в последовательность байтов;
- UnicodeTranslationError — ошибка преобразования строки в другую кодировку;
- ValueError — переданный параметр не соответствует ожидаемому значению;
- ZeroDivisionError — попытка деления на ноль.

Обработка исключений в Python

Инструкция try...except...else...finally.

- Для обработки исключений предназначена инструкция **try.**

  Формат инструкции:

```
try:
    <Блок, в котором перехватываются исключения>
[except [<Исключение1>[ as <Объект исключения:>]]:
    <Блок, выполняемый при возникновении исключения>
[ ...
except [<ИсключениеN>[ as <Объект исключения>]]:
    <Блок, выполняемый при возникновении исключения>]]
[else:
    <Блок, выполняемый, если исключение не возникло>]
[finally:
    <Блок, выполняемый в любом случае>]
```

- Инструкции, в которых перехватываются исключения, должны быть расположены внутри блока **try.** В блоке **except** в параметре <**Исключение>** указывается класс обрабатываемого исключения.

Обработка исключений в Python

Инструкция try...except...else...finally.

- Если в блоке **try** возникло исключение, управление передается блоку **except.**
- В случае если исключение не соответствует указанному классу, управление передается следующему блоку **except.**
- Если ни один блок **except** не соответствует исключению, или блоки **except** отсутствуют, то исключение «всплывает» к обработчику более высокого уровня.
- Если исключение в программе вообще нигде не обрабатывается, оно передается обработчику по умолчанию, который останавливает выполнение программы и выводит стандартную информацию об ошибке.
- Таким образом, в обработчике может присутствовать несколько блоков **except** с разными классами исключений.
- После обработки исключения управление передается инструкции, расположенной сразу после обработчика.

Обработка исключений в Python

Исключения при операциях с файлами.

- **BlockingIOError** — не удалось заблокировать объект (файл или поток ввода/вывода);
- **ConnectionError** — ошибка сетевого соединения, возникает при открытии файла по сети;
- **FileExistsError** — файл или каталог с заданным именем уже существуют;
- **FileNotFoundError** — файл или каталог с заданным именем не обнаружены;
- **InterruptedError** — файловая операция неожиданно прервана по какой-либо причине;
- **IsADirectoryError** — вместо пути к файлу указан путь к каталогу;
- **NotADirectoryError** — вместо пути к каталогу указан путь к файлу;
- **PermissionError** — отсутствуют права на доступ к указанному файлу или каталогу;
- **TimeoutError** — истекло время, отведенное системой на выполнение операции.

Обработка исключений в Python

Пример. Обработка исключений при операциях с файлами.

```
...
try:
    open("C:\temp\new\file", "r")
except FileNotFoundError:
    print("Файл отсутствует")
except IsADirectoryError:
    print("Это не файл, а каталог")
except PermissionError:
    print("Отсутствуют права на доступ к файлу")
except OSError:
    print("Неустановленная ошибка открытия файла")
...
```

ЦЕНТР
ДОПОЛНИТЕЛЬНОГО
ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

do.bmstu.ru