**Лекция 4.2. Валидация вводимых данных с использованием Python**

В данной лекции рассмотрены следующие вопросы:  
1. Необходимость валидации данных.  
2. Негативные последствия у отсутствия валидации.  
3. Способы валидации.

Глава 1. Валидация вводимых данных с использованием Python

Для чего нужна валидация данных?  
Казалось бы, «невалидные» данные, не удовлетворяющие определённым ограничениям, могут вызвать сбой в работе программы. Но что это означает? Предположим, в каком-то месте программы возникает исключение при попытке преобразовать строку в число, если строка имеет некорректный формат. Разумеется, если исключение не будет нигде перехвачено, это может привести к аварийному завершению программы. Но это маловероятный сценарий развития событий. Скорее всего в каком-то месте сработает перехватчик, который либо выдаст пользователю какое-то сообщение об ошибке в программе, либо сделает запись в журнал ошибок, после чего программа постарается восстановиться от сбоя и продолжить работу.  
Негативные последствия у отсутствия валидации:  
1. Невозможность восстановиться после сбоя.  
2. Дополнительная нагрузка на систему.  
3. Инъекции не вызывают сбоев.  
4. Сложность идентификации причины проблемы.  
Отсутствие валидации может приводить к вышеописанным (а может быть и ещё каким-то другим) проблемам. Соответственно, наличие валидации позволяет предотвратить серьёзные сбои, упрощает идентификацию проблем, но за это приходится расплачиваться производительностью, поскольку дополнительные проверки увеличивают нагрузку на систему. И тут мы переходим ко второму вопросу — как уменьшить эту дополнительную нагрузку.  
Удобно проверять данные в том месте, где они попадают в программу из внешнего мира. После такой проверки можно быть уверенным, что в программу попадают правильные данные и в дальнейшем они могут использоваться без дополнительных проверок.

Глава 2. Валидация вводимых данных с использованием Python

Способы валидации  
Где бы ни выполнялась валидация, можно это делать несколькими различными способами, в зависимости от того, какие ограничения накладываются на данные:  
1. Посимвольная проверка. Как правило, такие проверки выполняются в пользовательском интерфейсе, по мере ввода данных. Но не только. Например, лексический анализатор компилятора тоже выявляет недопустимые символы непосредственно в процессе чтения компилируемого файла. Поэтому такие проверки можно условно назвать «лексическими».  
2. Проверка отдельных значений. Для пользовательского интерфейса это проверка значения в отдельном поле, причём выполняться она может как по мере ввода (проверяется то неполное значение, которое введено к настоящему моменту), так и после завершения ввода, когда поле теряет фокус. Для программного интерфейса (API) это проверка одного из параметров, переданных в вызываемую процедуру. Для данных, получаемых из файла, это проверка какого-то прочитанного фрагмента файла. Такие проверки, опять-таки по аналогии с компиляторной терминологией, можно назвать «синтаксическими».  
3. Совокупность входных значений. Можно предположить, что в программу сначала передаются какие-то данные, после чего подаётся некоторый сигнал, который инициирует их обработку. Например, пользователь ввёл данные в форму или в несколько форм (в так называемом «визарде») и наконец нажал кнопку «OK». В этот момент можно выполнить так называемые «семантические» проверки, нацеленные на валидацию не только отдельных значений, но и взаимосвязей между ними, взаимных ограничений.  
4. Проверка состояния системы после обработки данных. Наконец, есть последний способ, к которому можно прибегнуть, если валидацию непосредственно входных данных выполнить не удаётся — можно попытаться их обработать, но оставить возможность вернуть всё к исходному состоянию. Такой механизм часто называется транзакционным.

1. Какой фактор необходимости в валидации данных предполагает предотвращение ошибок, которые могут возникнуть при вводе данных?
   1. Улучшение пользовательского опыта
   2. Безопасность данных
   3. Гарантирование целостности данных
   4. Предотвращение ошибок
2. Какой фактор необходимости в валидации данных связан с тем, что валидация позволяет проверить введённые данные на наличие потенциально опасных или вредоносных элементов, таких как SQL-инъекции или скрипты, и предотвращает их исполнение?
   1. Улучшение пользовательского опыта
   2. Безопасность данных
   3. Гарантирование целостности данных
   4. Предотвращение ошибок
3. Какой фактор необходимости в валидации данных связан с тем, что она позволяет убедиться в их правильности и соответствии ожидаемым значениям?
   1. Улучшение пользовательского опыта
   2. Безопасность данных
   3. Гарантирование целостности данных
   4. Предотвращение ошибок
4. Какой фактор необходимости в валидации данных связан с тем, что валидация данных помогает предупреждать пользователей об ошибках, предоставляя им информацию о правильном формате и требованиях ввода данных?
   1. Улучшение пользовательского опыта
   2. Безопасность данных
   3. Гарантирование целостности данных
   4. Предотвращение ошибок
5. Какое негативное последствие отсутствия валидации данных связано с тем, что если данные не проходят валидацию, то они могут быть некорректными или содержать ошибки?
   1. Непредвиденные ошибки и сбои
   2. Некорректные данные
   3. Потеря данных
   4. Нарушение безопасности
6. Какое негативное последствие отсутствия валидации данных связано с тем, что оно может привести к уязвимостям в системе?
   1. Непредвиденные ошибки и сбои
   2. Некорректные данные
   3. Потеря данных
   4. Нарушение безопасности
7. Какое негативное последствие отсутствия валидации данных связано с тем, что некорректные данные могут привести к потере или искажению информации?
   1. Непредвиденные ошибки и сбои
   2. Некорректные данные
   3. Потеря данных
   4. Нарушение безопасности
8. Какой способ валидации помогает удостовериться, что введённые данные соответствуют ожидаемому типу?
   1. Использование регулярных выражений
   2. Создание пользовательских функций
   3. Проверка типов данных
   4. Использование сторонних библиотек
   5. Использование условных операторов
9. Какой способ валидации позволяет использовать регулярные выражения для проверки правильности ввода электронной почты или номера телефона?
   1. Использование регулярных выражений
   2. Создание пользовательских функций
   3. Проверка типов данных
   4. Использование сторонних библиотек
   5. Использование условных операторов
10. При каком способе валидации мы можем проверить, является ли число положительным или отрицательным, и принять соответствующие меры?
    1. Использование регулярных выражений
    2. Создание пользовательских функций
    3. Проверка типов данных
    4. Использование сторонних библиотек
    5. Использование условных операторов