# Лабораторна робота 9 Обробка виключних ситуацій в Python

**Мета роботи:** Ознайомитись з технологією обробки виключних ситуацій в Python. Побудувати блок-схеми алгоритмів, розробити програми з використанням засобів обробки виключних ситуацій.

## Рейтинг лабораторної роботи №10

|  |  |  |  |
| --- | --- | --- | --- |
| № п.п | Вид діяльності студента | Рейтинговий бал | Deadline |
| 1 | Написання коду | 2 | грудень |
| 2 | Захист роботи | 1 |
| 3 | Звіт з роботи | 1 |
| 4 | Якість роботи (за наявності) | 1 |
| Разом за роботу мовою Python | | 5 |

## Теоретичний матеріал

Якщо у програмі виникає синтаксична помилка або помилка часу виконання, то відбувається **виключна ситуація**. Будь-яка виключна ситуація генерує (породжує) **виключення.**

Виключення (eng: exception) – спеціальний тип даних, що використовується для ідентифікації та детального опису помилки, що виникла у програмі.

Вбудовані виключення у мові Python організовані у вигляді ієрархії класів:

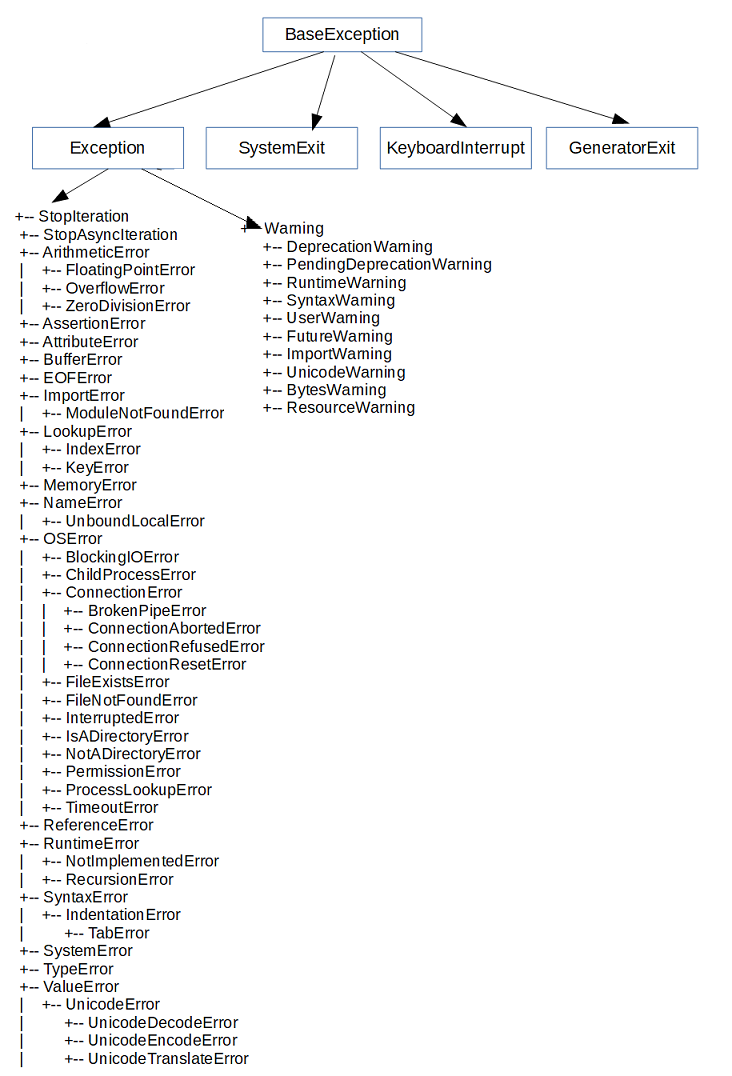


Рис. 1. Дерево класів виключень

### Обробка виключень

Для обробки виключень використовується оператор **try**.

**try**:

processing\_code #Инструкции

**except** exception\_1: #тип исключения

handling\_exception\_1 #Что делать, например, закрыть файл

**except**:

handling\_exception\_2 #Что делать при любом ещё не пойманном

#исключении в этом блоке

**else**:

state\_else #Что делать, если не было исключений

**finally**:

obligatory\_code #Что делать в любом случае после блока

**Виконання цього оператора відбувається таким чином:**

1. Інтерпретатор намагається виконати блок коду з потенційним виключенням (processing\_code).
2. Якщо у випадку виконання коду породжується виключення, що належить до класу виключень exception\_1 то у точці генерування виключення відбувається безумовний перехід на блок обробки виключення handling\_exception\_1. Аналогічне стосується інших типів виключення, що перехоплюються у програмі.
3. Якщо, породжене виключення не обробляється у жодному з блоків обробки виключень, то інтерпретатор припиняє роботу програми (за допомогою типового обробника виключень)
4. Якщо виконання блоку **try** повністю відбулося без виключень, то відбувається виконання коду state\_else блоку **else**. Якщо ж було згенеровано хоча б одне виключення цей блок не виконується.
5. Код obligatory\_code блоку **finaly** виконується завжди після виконання попередніх кроків незалежно від того, чи було породжено виключення чи ні.

Обов’язковими блоками оператора **try** є лише блок **try** і один блок **except** обробки виключень або блок **try** і блок **finally**

**Приклад.** Створити масив чисел, які вводять з клавіатури. Здійснити контроль введення через обробку виключних ситуацій. Знайти суму елементів масиву. Кількість елементів визначається в процесі вводу. Якщо користувач вводить не числове значення, то генерується виключення.

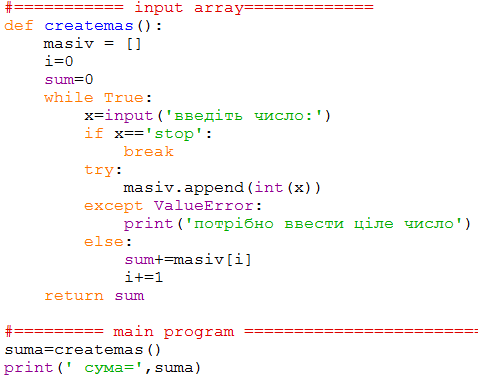


Рис. 2. Приклад коду

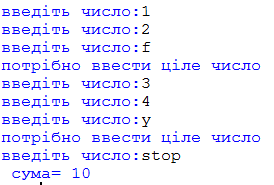


Рис. 3. Результати роботи програми

## Завдання для самостійної роботи за варіантами

Програму реалізувати мовою Python за завданнями лабораторної роботи №7 (а). Додати до коду лабораторної роботи №7 (а) обробку виключних ситуацій відповідно до завдання.

Для підвищення рейтингу написати програму лабораторної роботи №7 (а) мовою С без обробки виключних ситуацій. Кількість додаткових балів за код мовою С дорівнює 5.

### Варіанти завдань

1. **Скласти програму для обробки одновимірного масиву. Передбачити обробку виключних ситуацій**

Ввести список чисел з клавіатури. Елементи списку є додатними, від’ємними та рівними нулю числами. З введеного списку сформувати новий список, взявши в нього тільки ті елементи із вхідного списку, які більше за модулем введеного з клавіатури числа M. Вивести на екран число M, вхідний та отриманий списки. Передбачити виключні ситуації: нечислового символу, відсутність від’ємних та нульових чисел.

1. **Скласти програму для обробки одновимірного масиву. Передбачити** введення **обробку виключних ситуацій**

Створити список чисел, кількість елементів якого задана користувачем. Визначити максимальне та мінімальне значення серед елементів із парними та непарними індексами. Вивести список, мінімальний, максимальний елементи та їх індекси. Передбачити виключні ситуації: введення не додатної кількості елементів масиву та нечислових значень елементів масиву.

1. **Скласти програму для обробки одновимірного масиву. Передбачити обробку виключних ситуацій**

Створити числовий список, кількість елементів якого задана користувачем. Обчислити суму елементів між максимальним та мінімальними значеннями списку. Вивести на екран список, максимальний та мінімальний елементи масиву та їх індекси. Передбачити виключні ситуації: введення нечислових значень елементів масиву, вихід індексів елементів за межі масиву.

1. **Скласти програму для обробки одновимірного масиву. Передбачити обробку виключних ситуацій.**

Створити два числових списки, кількість елементів яких задана користувачем. Знайти найменший серед тих елементів першого списку, які співпадають із значеннями елементів другого списку. Вивести на екран списки, найменший елемент та його індекс. Передбачити виключні ситуації: введення нечислових значень елементів масиву, відсутність однакових елементів в двох списках.

1. **Скласти програму для обробки одновимірного масиву. Передбачити обробку виключних ситуацій.**

Створити список цілих чисел. Побудувати новий список із елементів першого списку. В новому списку спочатку стоять числа, що діляться на 2, потім ті, що діляться на 2 та 3, потім на 3. Надрукувати вхідний та вихідний списки. Передбачити виключні ситуації: введення нечислових значень елементів масиву, відсутність елементів, які кратні 2 та 3.

1. **Скласти програму для обробки одновимірного масиву. Передбачити обробку виключних ситуацій.**

Створити список. Обчислити кількість елементів у найдовшій серії. Серія — це послідовність однакових елементів, розташованих поспіль. Надрукувати список, усі серії однакових елементів та довжину найдовшої серії. Передбачити виключні ситуації: введення нечислових значень елементів масиву, відсутність послідовностей однакових елементів.

1. **Скласти програму для обробки одновимірного масиву. Передбачити обробку виключних ситуацій.**

У згенерованому числовому списку знайти найбільший серед від’ємних та найменший серед додатних елементів списку. Вивести список, значення знайдених елементів та їх індекси. Передбачити виключні ситуації: введення некоректної (нульової, від’ємної, нечислової) вимірності масиву, відсутність від’ємних елементів в масиві.

1. **Скласти програму для обробки одновимірного масиву. Передбачити обробку виключних ситуацій.**

Згенерувати список з цілих чисел та вивести його на екран. Знайти максимальний елемент та його індекс та надрукувати їх. Вивести на екран кількість елементів, які менші за значення максимального. Передбачити виключні ситуації: введення некоректної (нульової, від’ємної, нечислової) вимірності масиву та вихід індексу за межі масиву.

1. **Скласти програму для обробки одновимірного масиву. Передбачити обробку виключних ситуацій.**

Згенерувати два упорядкованих за зростанням числових списків. Побудувати упорядкований за зростанням третій список злиттям двох згенерованих. Передбачити виключні ситуації: введення некоректної (нульової, від’ємної, нечислової) вимірності масиву та вихід індексу за межі масиву.

1. **Скласти програму для обробки одновимірного масиву. Передбачити обробку виключних ситуацій.**

Ввести з клавіатури два списки цілих чисел. Кількість елементів списків вводиться з клавіатури. Побудувати третій список, в якому кожний елемент дорівнює найбільшому спільному дільнику відповідних елементів введених списків. Використати алгоритм Евкліда. Надрукувати усі три списки. Передбачити виключні ситуації: введення некоректної (нульової, від’ємної, нечислової) вимірності масиву, невідповідність двох вихідних списків.

1. **Скласти програму для обробки одновимірного масиву. Передбачити обробку виключних ситуацій.**

Створити числовий список, кількість елементів якого вводиться користувачем. Значення елементів в списку можуть повторюватися. Визначити індекс першого та останнього входження заданого числа в список. Знайти елемент, значення якого повторюється найбільшу кількість разів. Вивести на екран список, шуканий елемент та його індекс. Передбачити виключні ситуації: введення некоректної (нульової, від’ємної, нечислової) вимірності масиву, вихід індексів за межі масиву.

1. **Скласти програму для обробки одновимірного масиву. Передбачити обробку виключних ситуацій.**

Створити список, значеннями елементів якого є коефіцієнти многочлена I:\!TeachingStudentKPI2018\course1_A&P\A&Psemestr1\LabA&P_semestr1\MethodLabWorkC++Semestr1Html\labs\lab_07\pictures\var11.gif. Знайти значення многочлена при введеному з клавіатури аргументу *x* та похідної від многочлена *A*(*x*) при заданому *x*. Вивести на екран список, значення многочлена та його похідної. Передбачити виключні ситуації: введення нечислових значень коефіцієнтів многочлена та аргументу *х*

1. **Скласти програму для обробки одновимірного масиву. Передбачити обробку виключних ситуацій.**

Згенерувати список цілих чисел, кількість елементів якого ввести з клавіатури. Вивести список на екран. Отримати інший список, що складається тільки з непарних чисел вихідного списку або повідомити, що таких чисел немає. Отриманий список вивести на екран в порядку спадання елементів. Передбачити виключні ситуації: введення нечислового, нульового, від’ємного значення кількості елементів списку, відсутність непарних чисел в масиві.

1. **Скласти програму для обробки одновимірного масиву. Передбачити обробку виключних ситуацій.**

Ввести з клавіатури список цілих чисел. Знайти в списку все прості числа і скопіювати їх в новий список. Надрукувати вхідний та вихідний списки. Передбачити виключні ситуації: некоректність введених даних, відсутність простих чисел.

1. **Скласти програму для обробки одновимірного масиву. Передбачити обробку виключних ситуацій.**

Згенерувати список чисел, кількість елементів якого ввести з клавіатури. У списку чисел перевірити, чи всі елементи є унікальними, тобто кожне число зустрічається тільки один раз. Надрукувати згенерований список та повідомлення «Усі елементи унікальні» або «Не усі елементи унікальні», які сформувати в результаті обробки виключних ситуацій.

1. **Скласти програму для обробки одновимірного масиву. Передбачити обробку виключних ситуацій.**

Згенерувати числовий список, кількість елементів якого ввести з клавіатури. Визначити індекси елементів списку, значення яких не менше заданого мінімуму і не більше заданого максимуму. Вивести на екран список, індекси та значення елементів, що належать заданому діапазону. Передбачити виключні ситуації: некоректне введення кількості елементів масиву, відсутність елементів в заданому діапазоні.

1. **Скласти програму для обробки одновимірного масиву. Передбачити обробку виключних ситуацій.**

Згенерувати числовий список з елементами від -100 до +100. Надрукувати список. Вивести послідовність елементів між першим від’ємним і нульовим елементом, між першим додатним і нульовим елементом. Передбачити виключні ситуації: відсутність елементів масиву , що відповідають заданим діапазонам.

1. **Скласти програму для обробки одновимірного масиву. Передбачити обробку виключних ситуацій.**

Ввести з клавіатури числовий список з трьох елементів. У кожного елемента списку є вага, яка збільшується на одиницю, коли користувач вибирає цей елемент списку. Початкове значення ваг дорівнює нулю. Користувач має *n* спроб вибрати кожний елемент списку. Вибір елементів здійснюється випадково за допомогою генератора псевдовипадкових чисел. При випаданні потрібного елемента, його вага збільшується. Вивести на екран значення ваг елементів після *n* спроб вибрати їх. Передбачити виключні ситуації: введення нечислових значень списку, генератор не генерує числа, які є в масиві.

1. **Скласти програму для обробки одновимірного масиву. Передбачити обробку виключних ситуацій.**

Ввести з клавіатури список чисел. Визначити, скільки в цьому списку елементів, які більше двох своїх сусідів справа та зліва. Вивести кількість таких елементів. Крайні елементи списку не враховуються, оскільки у них недостатньо сусідів. Передбачити виключні ситуації: введення нечислових значень списку, відсутність елементів, які більше двох своїх сусідів справа та зліва

1. **Скласти програму для обробки одновимірного масиву. Передбачити обробку виключних ситуацій.**

Згенерувати список чисел, кількість елементів якого ввести з клавіатури. Вивести згенерований список. Обчислити кількість пар елементів, рівних один одному. Вважається, що будь-які два елементи, які рівні один одному, утворюють одну пару. Вивести індекси елементів, що утворюють пару та кількість пар. Передбачити виключні ситуації: введення нечислового значення кількості елементів масиву, відсутність пар елементів, рівних один одному.

1. **Скласти програму для обробки одновимірного масиву. Передбачити обробку виключних ситуацій.**

Згенерувати список цілих додатних та від’ємних чисел, кількість елементів якого ввести з клавіатури. Знайти суму елементів, розташованих до першого від’ємного елементу та добуток елементів, розташованих після останнього від’ємного елемента. Передбачити виключні ситуації: введення нечислового значення кількості елементів масиву, відсутність від’ємних елементів.

1. **Скласти програму для обробки одновимірного масиву. Передбачити обробку виключних ситуацій.**

Ввести з клавіатури список чисел. Видалити зі списку всі числа, які повторюються більше двох разів. Надрукувати вхідний список і список після видалення елементів. Передбачити виключні ситуації: введення нечислових значень, відсутність чисел, які повторюються більше двох разів в масиві.

1. **Скласти програму для обробки одновимірного масиву. Передбачити обробку виключних ситуацій.**

Згенерувати список цілих чисел в діапазоні від -50 до +50, кількість елементів задати з клавіатури. Із отриманого списку скопіювати в інший список елементи, значення яких знаходяться в діапазоні від -5 до 5 включно і підрахувати їх кількість. Надрукувати згенерований і побудований списки. Передбачити виключні ситуації: відсутність елементів, значення яких знаходяться в діапазоні від -5 до 5.

1. **Скласти програму для обробки одновимірного масиву. Передбачити обробку виключних ситуацій.**

Дан список цілих чисел. Розмір списку і його елементи вводяться користувачем. Упорядкувати за зростанням елементи списку, що розташовані до максимального елемента, і за спаданням – ті елементи, що розташовані після максимального елементу. Надрукувати список до та після сортування. Передбачити виключні ситуації: введення нечислових значень розміру та елементів списку, вихід індексів за межі масиву

1. **Скласти програму для обробки одновимірного масиву. Передбачити обробку виключних ситуацій.**

Дан список дійсних чисел. Розмір списку і значення його елементів вводяться користувачем. Знайти суму всіх локальних мінімумів списку. Локальний мінімум – це елемент списку, який менший за значенням двох сусідніх елементів даного списку (виключаючи крайні елементи). Надрукувати вхідний список, всі його локальні мінімуми та їх індекси. Передбачити виключні ситуації: введення нечислових значень розміру та елементів списку та відсутність локальних мінімумів.

## Список літератури

1. <https://github.com/tkovalyuk/Basics-of-programming>
2. **МакГрат М**. Программирование на Python для начинающих. –Москва: Эксмо. – 192 с.
3. **Лутц М.** Изучаем Python, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 1280 с.
4. **Лутц М.** Программирование на Python, том I, 2. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 992 с.
5. **Васильев А. Н.** Python на примерах. Практический курс ·по программированию. - СПб.:Наука и Техника, 2016. - 432 с.
6. **Доусон М.** Программируем на Python. - СПб.: Питер, 2014. - 416 с.
7. **Рейтц К., Шлюссер Т.** Автостопом по Python. — СПб.: Питер, 2017. — 336 с.