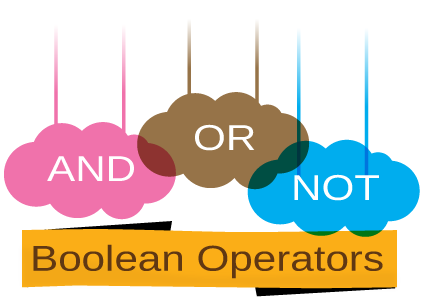
**Lesson 11. Boolean operators**

У даному занятті маємо дві головні цілі. Перша – познайомитись із логічними операторами в Java. Друга – познайомитись із константами в Java.

Обидві теми не надто об’ємні, але від того не менш важливі, ніж будь-які інші.

Крім того, ми далі продовжимо працювати над обробкою текстових даних, а завдяки новим темам зможемо зробити наш код більш ефективним та зрозумілим.

**Boolean operators**

Оператори порівняння дозволяють створювати прості умови, які складаються із одного порівняння. Однак іноді необхідно задавати умови, які називають **складними**, тобто такі, що складаються з двох і більше простих умов.

В українській мові, так само як і Java, також зустрічаються прості і складні умови:

* Проста умови: «*Якщо буде гарна погода, ми підемо на прогулянку*».
* Складна умови: «*Якщо буде гарна погода і ти виконаєш домашнє завдання, ми підемо на прогулянку*».

У останньому прикладі складну умову утворює сполучник «і». Такий самий сполучник можемо застосувати для складання двох умов пов’язаних із числами і операторами порівняння:

«*Якщо 5 > 2 і 5 < 10, то я запам’ятаю число 5».*

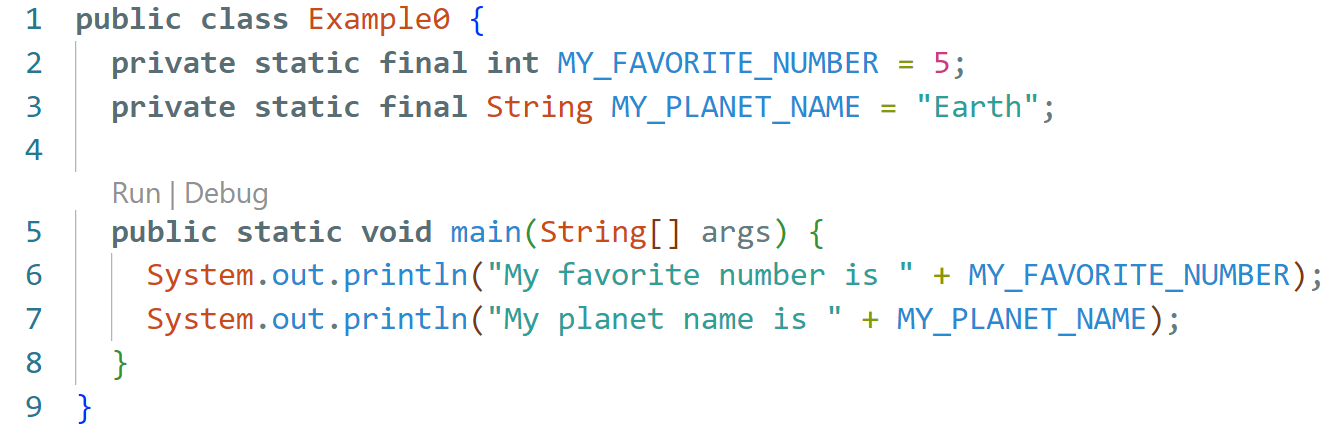
У мові Java роль таких сполучників виконують логічні оператори. Опис і приклади використання наведені в таблиці нижче:

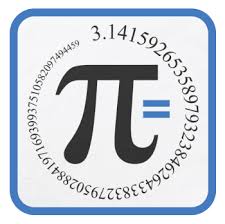
|  |  |  |  |
| --- | --- | --- | --- |
| Оператор | Назва | Опис | Приклад |
| && | Логічне «І» (AND) | Якщо поєднані логічні вирази обидва мають значення true, то і складений матиме значення true, а інакше – false. | a > b && a <c |
| ! | Логічне заперечення (NOT) | Якщо логічний вираз true, то результатом буде false, якщо false – навпаки true. | !(a < b) |
| || | Логічне «АБО» (OR) | Якщо хоча б один із поєднаних логічних виразів має значення true, то і складений вираз матиме значення true, а інакше – false. | a > b || a <c |

**Константи** – це (якщо коротко) змінні, значення яких не можна змінити. От такий от парадокс, «змінні, які не можна змінювати».

**Constants**

Значення константи надається їй тільки раз при створенні. Отже звідси і перше питання: «як створити константу?». Для цього в Java використовуються модифікатори static і final. Наприклад:



Зверніть увагу, що оголошення констант відбувається поза методом main. Що означають і як працюють дані модифікатори на даний момент ми детально розбирати не будемо, але обов’язково з’ясуємо на подальших заняттях.

У константах варто зберігати саме ту інформацію, яка не може і не повинна бути змінена під час роботи програми. Детальніше розглянемо під час виконання практичних завдань.

**Examples**

**Приклад 1**. Наступна програма повинна приймати від користувача username і перевіряти, чи він відповідає вимогам по кількості символів. Username повинен бути мінімум 4 символи довжино і максимум – 50 символів.

Тестові дані:

*Enter your username: Nik*

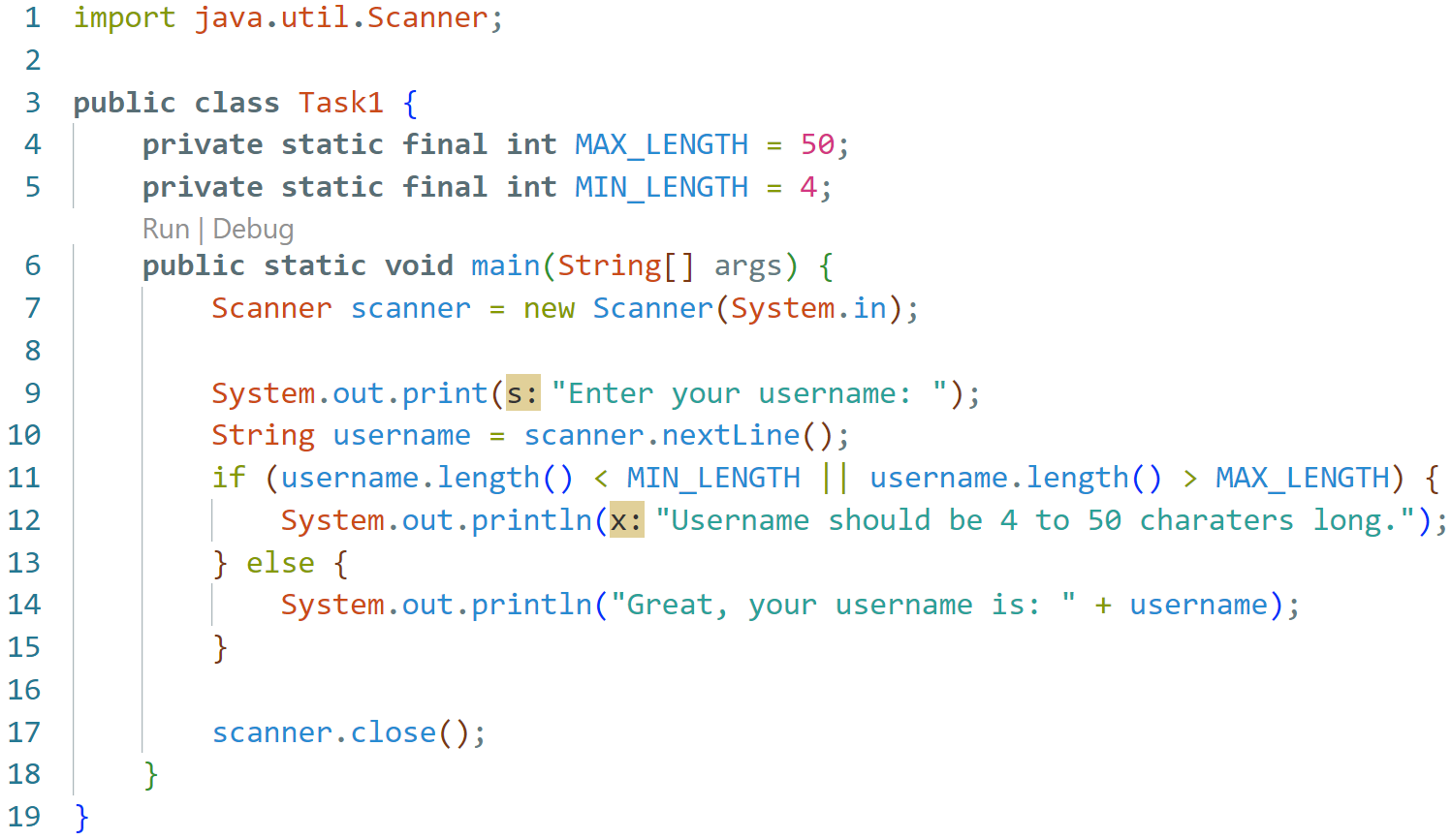
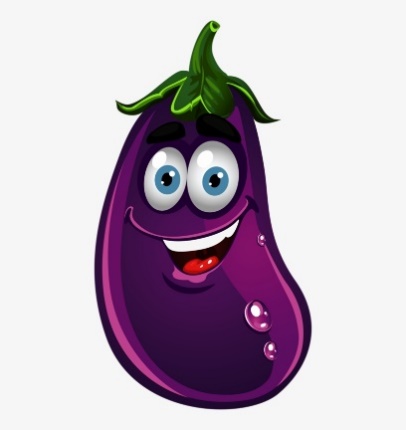
*Enter your username: Nikson*

Очікуваний результат:

*Username should be 4 to 50 charaters long.*

*Great, your username is: Nikson*

Проаналізуйте наведені приклади, та зробіть висновок про те, що спільного між ним і чим принципово вони відрізняються. Визначте, чи є вони взаємозамінними для вирішення даної задачі.

**Завдання 1**. Є така математична гра, у якій гравці по колу називають числа по порядку від 1. Числа, що діляться на 3 без остачі, або які мають у своєму складі цифру 3 необхідно замінювати спеціальним словом (наприклад: «баклажан»).

**Tasks**

Створіть програму, що буде отримувати від користувача одне число і визначатиме, чи потрібно його було б змінити у нашій грі на спеціальне слово.



Тестові дані:

*Enter number: 20*

*Enter number: 27*

*Enter number: 23*

Очікуваний результат:

*20*

*Aubergine*

*Aubergine*

**Завдання 2**. Створіть програму, що буде приводити номер телефона введений користувачем до «гарного вигляду».

Тестові дані:

*Enter your phone number: 380502726343*

*Enter your phone number: 0502726343*

Очікуваний результат:

*Your number is: +38 (050) 272-63-43*

*Phone number should be 12cahracters long.*

**Завдання 3**. Створіть програму, що буде із фрази введеної користувачем створювати абревіатуру. Слова у фразі повинні бути розділені пробілами:

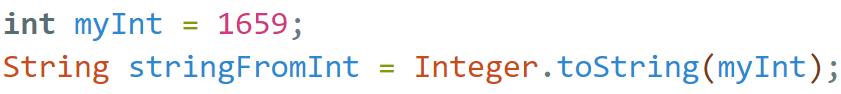
**New things**

Тестові дані:

*Enter your phrase: National Football League*

Очікуваний результат:

*NFL*

1. Для перетворення значень різних типів у String існують методи toString(). Вони належать класам-обгорткам примітивних типів. Що воно таке ті обгортки знати зараз не обов’язково, але обов’язково знати як перетворювати примітивні типи у значення String:

Такі ж самі методи містять інші класи Character, Long, Boolean та інші.

1. Ще один корисний метод із класу String – **substring**(beginIndex, endIndex). Він дозволяє відрізати із рядка певну частину за вказаними початковим і кінцевим індексом. Наприклад:



Результатом виведення даного коду буде текст: **ootba**. Зверніть увагу, що endIndex – значення, яке не включається у проміжок, що вирізається із рядка.

**Homework**

**Завдання 1.** Клініка JavaMed має двох сімейних лікарів. Лікар Вовк приймає з 9:00 до 12:00, а лікарка Гринь – з 16:00 до 19:00.

Завдання полягає у тому, щоб зробити програму, яка буде підбирати лікаря за часом візиту. Тобто користувач вводить час, на який би він хотів записатись до лікаря, а програма видає який лікар зможе його прийняти.

Тестові дані:

*Enter an hour when you want to start visit: 18*

*Enter an hour when you want to start visit: 10*

*Enter an hour when you want to start visit: 8*

Очікуваний результат:

*Dr. Gryn will be ready for your visit.*

*Dr. Vovk will be ready for your visit.*

*Sorry, but there are no doctors working at this time.*