

Завдання 3 — Problem-solving in Java

Ось ми практично пройшли вивчення фундаменту імперативного програмування — цикли, змінні, методи, параметри і можливість повертати значення із методів. Час тепер це попрактикувати!

Ваша мета: заімплементити декілька маленьких програмок, ну, як зазвичай. Закріпити свій фундамент, так би мовити.

Частина 1 — Аеробіка

Медики рекомендують 30 хвилин занять аеробіки п'ять днів на тиждень (тут було написано: з пн по пт, але в цілому мій план провалився) для підтримання свого кардіоваскулярного здоров'я. Більш того, хоча б 40 хвилин три рази на тиждень для підтримання низького кров'яного тиску. Ваше завдання: написати програму, яка запитує у користувача кількість хвилин, потрачених останні сім днів на вправи, і відповідно, репортить наступне:

- чи було достатньо часу приділено вправам для кардіоваскулярного здоров'я, і, якщо ні, виводить: скільки днів правильного життя (с заняттями більше 30 хвилин) не вистачає для рекомендованого графіку.
- чи було достатньо вправ для зменшення кров'яного тиску і холестерину, і, якщо ні, виводить: скільки днів користувач не дотягнув з заняттями по 40 хвилин на день.

```
How many minutes did you do on day 1? 40
How many minutes did you do on day 2? 30
How many minutes did you do on day 3? 30
How many minutes did you do on day 4? 30
How many minutes did you do on day 5? 11
How many minutes did you do on day 6? 40
How many minutes did you do on day 7? 30
```

Cardiovascular health:

Great job! You've done enough exercise for cardiovascular health.

Blood pressure:

You needed to train hard for at least 1 more day(s) a week!

How many minutes did you do on day 1? 15

How many minutes did you do on day 2? 60

How many minutes did you do on day 3? 15

How many minutes did you do on day 4? 60

How many minutes did you do on day 5? 15

How many minutes did you do on day 6? 60

How many minutes did you do on day 7? 15

Cardiovascular health:

You needed to train hard for at least 2 more day(s) a week!

Blood pressure:

Great job! You've done enough exercise to keep a low blood pressure.

Частина 2 — Числа-градини 😊

В книзі Дугласа Хофштадтера “Гедель, Ешер, Бах” (відзначена Пулітцерівською премією) згадуються різні цікаві завдання, які, якоюсь мірою, можуть бути розглянуті через призму програмування. Ось одна із них:

Взяти якесь додатне ціле число і назвати його n

Якщо n - парне, то поділити його на 2

Якщо n - непарне, то помножити на 3 і додати 1

Продовжувати цей процес до тих пір поки n не дорівнюватиме 1

Ось, наприклад, процес для числа 15:

15 - непарне, значить потрібно помножити на 3 и додати 1, отримаємо 46

46 - парне, значить потрібно поділити на 2, отримаємо 23

23 - непарне, значить потрібно помножити на 3 и додати 1,

```
отримаємо 70
70 - парне, значить потрібно поділити на 2, отримаємо 35
35 - непарне, значить потрібно помножити на 3 и додати 1,
отримаємо 106
106 - парне, значить потрібно поділити на 2, отримаємо 53
53 - непарне, значить потрібно помножити на 3 и додати 1,
отримаємо 160
160 - парне, значить потрібно поділити на 2, отримаємо 80
80 - парне, значить потрібно поділити на 2, отримаємо 40
40 - парне, значить потрібно поділити на 2, отримаємо 20
20 - парне, значить потрібно поділити на 2, отримаємо 10
10 - парне, значить потрібно поділити на 2, отримаємо 5
5 - непарне, значить потрібно помножити на 3 и додати 1,
отримаємо 16
16 - парне, значить потрібно поділити на 2, отримаємо 8
8 - парне, значить потрібно поділити на 2, отримаємо 4
4 - парне, значить потрібно поділити на 2, отримаємо 2
2 - парне, значить потрібно поділити на 2, отримаємо 1
кінець.
```

Таким чином, ми бачимо злети і падіння, але все приходить до 1. Це схоже на те, як град тримається в хмарах, але в якийсь момент падає на землю, тому і називається ця послідовність “Числами-градинами”.

Отже, вам потрібно написати програму, яка зчитує у користувача число і видає схоже на це:

```
Enter a number: 17
17 is odd so I make 3n + 1: 52
52 is even so I take half: 26
...
```

Що цікаво, ніхто не довів, що може існувати число, яке не призведе до 1 в результаті, з іншого боку немає доказів зворотного.

Частина 3 — Піднесення до степеня

Ваше завдання — написати метод:

```
private double raiseToPower(double base, int exponent)
```

Цей метод буде приймати два параметри і обчислювати значення першого параметра, піднесеного до степеня параметра 2. тобто `raiseToPower(2.0, 3)` дасть $2^3 = 8$. Другий параметр може бути не тільки додатнім, але і від'ємним. Наприклад `0.5 в степені -2` це 4. Для спрощення вашого життя вважатимемо, що все, що підноситься до степеня 0, стане 1, тобто `0 в степені 0` буде 1 😊

Оскільки `double` не вміє зберігати в собі нескінченно довгі дробові числа (як ми обговорювали), то при маніпуляції з даблами можливі всілякі дрібні похибки. Це ок.

Також, вам не дозволено в цьому підзавданні використовувати `Math.pow`, `Math.exp`, `Math.log` та інші зручні методи із `Math`, інакше який тоді сенс, в написанні одного метода, чи не так? 😊

Частина 4 — Піраміда

Отже, потрібно буде забацати пірамідку із цеглинок. Кожен рядок містить на 1 цеглинку менше. Піраміда має бути відцентрована по горизонталі та лежати на “дні” вікна. Константи — `BRICK_HEIGHT`, `BRICK_WIDTH`, `BRICKS_IN_BASE` (скільки цеглинок в основі)

Частина 5 — Санкт-Петербурзька гра

Це гіпотетична гра для казино з простою ідеологією. Грають дві людини: щасливчик і спітнілий. Гра закінчується, коли перший заробляє **\$20 або більше**. Спітнілий кладе \$1 на стіл, а щасливчик починає підкидати монетку. *Якщо орел* — то спітнілий докладає до суми на столі точно таку ж суму. *Якщо решка* — все, що на столі, переходить до щасливчика. Якщо у щасливчика в результаті **менше ніж \$20, то гра повторюється**.

Теоретично, навіть перша гра може призвести до того, що щасливчику замість 20 привалить одразу \$32, \$64, ... в залежності від того, наскільки йому буде щастить з орлами.

Тобто, теоретично, на одній глобальній ітерації можна заробити нескінченну суму грошей. З іншого боку, може виявитись, що буде часто випадати решка, і сума всіх перемог складе ті самі \$20 (або близько до того).

Програма повинна імітувати подібну гру і виводити на екран наступне:

```
This game, you earned $1
Your total is $1
This game, you earned $1
Your total is $2
This game, you earned $1
Your total is $3
This game, you earned $128
Your total is $131
It took 4 games to earn $20
```

Втім, це потрібно бути дуже везучим, щоб дістатися до суми хоча б \$ 10000 ...

Частина 6 — П'ять секунд слави

Ми, можливо, ще не розглядали анімації, але скоро розглянемо. Ну, а поки що вам завдання: написати програму, яка виводить на екран цілих 5 секунд будь-якої анімації (не взятої звідкись, а власноруч створеної). Це може бути будь-яка анімація на ваш вибір з будь-яким сценарієм.

- п'ять секунд, не більше і не менше. В кінці нашого курсу ми об'єднаємо всі ваші творіння в один кльовий ролик.
- анімація повинна містити 50 фреймів принаймні. Якщо ж у вас 24 фрейми в секунду, то взагалі добре.

Проявіть свою креативність!) це — все, що ми просимо.

(опціонально) Частина 7 — Доповнення!

Якщо у вас є ідеї як доповнити вищевказані завдання новими свистілками та гуділками то зробіть це, це завжди вам плюс в карму)

Найменування файлів та додатків подібно до минулого завдання, **будь ласка дотримуйтеся правильного найменування файлів!**

Корисні поради

В завданні 2 було багато констант. Тут же часто йтиме робота з користувачем. Тестуйте свої програми з різними вхідними даними, оскільки буде сумно бачити хорошу роботу при стандартних вхідних

числах і фейл програми при якомусь незвичному значенні, введеному користувачем.

Не забувайте і про стиль, і про декомпозицію. Також, гарною ідеєю буде вводити свої константи, про які ми навіть не просили.

Сподіваємось вам подобається навчатись в Ш++. Ми існуємо завдяки вдячним студентам та небайдужим людям. Підтримайте нас, якщо хочете щоб ми продовжували вчити якісніше і більше людей 📧

Copyright 2022 Stanford University. Перекладено та доповнено Школою програмування Ш++.

2022 Made with by Ш++

