



Основні методи класу String:

int length()

- повертає довжину рядка

char charAt(**int** index)

- повертає символ за вказаним індексом (нумерація з 0)

boolean equals(String r)

- повертає *true* тоді й лише тоді, коли цей рядок складається з такої самої послідовності символів, як і рядок, на який посилається параметр r

String trim()

- повертає посилання на новий рядок без пробілів на початку та в кінці

String toLowerCase() / String toUpperCase()

- повертає посилання на новий рядок із малими / великими літерами

String substring(**int** zacIndex, **int** konIndex)

- повертає підрядок із символів між індексами zacIndex (включно) і konIndex (виключно)

int indexOf(String podretazec) або **int** indexOf(**char** znak)

- повертає індекс першої появи підрядка або символу, або -1, якщо немає

Основні методи класу Turtle:

void center()

- переміщує черепашку в центр полотна (черепаха мусить бути в полотні)

void setPosition(**double** x, **double** y)

- переміщує черепашку в точку з координатами [x, y]; лінія при цьому не малюється

void step(**double** dlzka)

- робить крок у поточному напрямку на задану довжину; малює, якщо перо опущене.

void turn(**double** uhol)

- повертає черепашку на заданий кут у напрямку годинникової стрілки

void moveTo(**double** x, **double** y)

- переміщує черепашку в точку [x, y]; малює, якщо перо опущене

void setDirection(**double** smer)

- встановлює напрямок черепашки (напрямок 0 — вгору, 90 — праворуч тощо)

double getDirection()

- повертає напрямок поточного обертання черепашки

void turnTowards(**double** x, **double** y)

- повертає черепашку так, щоб вона була спрямована на точку з координатами [x, y]

double directionTowards(**double** x, **double** y)

- повертає кут, під яким черепашка мала бути би обернена до точки [x, y]

double distanceTo(**double** x, **double** y)

- повертає відстань від черепашки до точки з координатами [x, y]

void dot(**double** polomer)

- малює заповнене коло заданого радіуса в поточній позиції

void setFillColor(Color farba)

- встановлює колір заливки

void setPenColor(Color farba)

- встановлює колір пера

void penDown() або **void** penUp()

- увімкнути або вимкнути малювання (опустити чи підняти перо)

void openPolygon()

- увімкнути відстеження меж області, яка буде залита після `void closePolygon()`

Основні методи класу WinPane (kresliaca plocha):

`void add(Turtle korytnacka)`

- додає черепашку на кресляче полотно

`void remove(Turtle korytnacka)`

- видаляє (згадану) черепашку з полотна

`int getWidth()` resp. `int getHeight()`

- повертає ширину або висоту креслячого полотна

Масиви в Java

- перебір усіх індексів масиву, на який посилається змінна *pole*:

```
for (int i=0; i<pole.length; i++) { ... }
```

JPAZ і події миші

```
protected void onMouseClicked(int x, int y, MouseEvent detail) {
    if ((detail.getButton() == MouseEvent.BUTTON1) &&
        detail.isControlDown()) {
        // при натисканні лівої кнопки миші,
        // коли одночасно натиснуто клавішу Ctrl
    }
}
```

Кольори

`Color.red`, `Color.blue`, `Color.green`, `Color.gray`, `Color.black` ... або
`new Color(int r, int g, int b)`, де *r*, *g* та *b* — цілі числа від 0 до 255.

Випадкові числа

Генерування випадкового числа з інтервалу $<0, a$): `Math.random()*a`

Генерування випадкового цілого числа від 0 до *n*: `(int)(Math.random()*(n+1))`

Створення масиву

Створення масиву з 6 цілих чисел: `int[] pole = new int[6];` Створення з початковими значеннями:
`int[] pole = {3, 4, 6, 1, 2, 4};`

Виведення: `System.out.println(Arrays.toString(pole));`

Копіювання елементів масиву:

```
System.arraycopy(звідки, зІндексу, куди, зІндексу, кількість);
```

Числа

`Double.MAX_VALUE` - найбільше число, яке можна зберегти у змінній типу `double`

`Double.POSITIVE_INFINITY` - $+\infty$

`double cislo = Double.parseDouble("3.14");` - перетворює рядок у число

`Math.sqrt(c)` - квадратний корінь числа *c*

Символи

`Character.isUpperCase(z)` - повертає `true`, якщо символ *z* є великою літерою

`Character.toUpperCase(z)` - повертає символ *z* у верхньому регістрі

