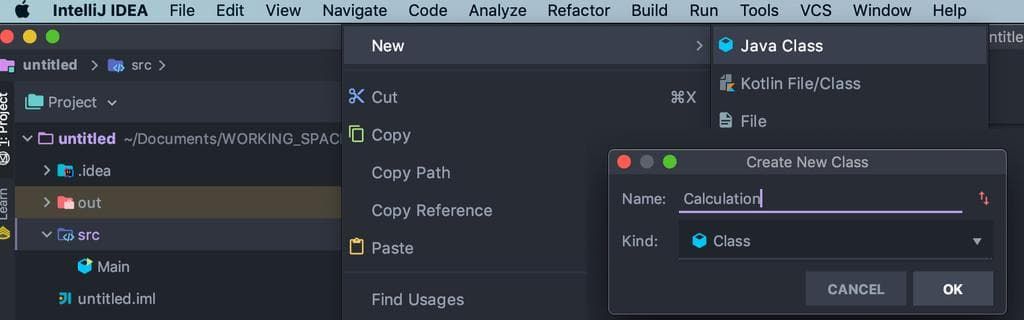
**Biến & Kiểu dữ liệu**

Chúng ta sẽ chạy một ví dụ trước rồi đi vào chi tiết nhé. Các bạn tạo một project mới, đặt tên là Bài 2 hay gì cũng được, tuỳ bạn chọn nhé. Và tạo cho mình một file là Calculation.java như hình.



Sau đó các bạn viết code như này và chạy thử:

public class Calculation{

public static void main(String[] args){

// khai bao so nguyen

int a = 5;

int b = 10;

int x = 10 + 5;

System.out.println(x);

}

}

Các bạn chạy chương trình này (click chuột phải vào file Calculation > Run Main()). sẽ thấy hiện kết quả là c = 15;

Nhìn code thì rất dễ hiểu phải không, tuy nhiên chúng ta cùng lí giải chi tiết để hiểu hơn về Biến và Kiểu dữ liệu trong Java.

Thứ nhất là cái // khai bao so nguyen, cái này gọi là Comment, tức các bạn viết gì sau 2 cái dấu // thì nó sẽ không ảnh hưởng tới code của chương trình, nó chỉ mang ý nghĩa chú thích thôi.

Thứ hai là cái này:

int a = 5;

Nói về Biến (Variable) các bạn có liên tưởng tới liên tưởng tới biến x trong đồ thị hàm số ax + b = 0 không 😂 Thì chính là nó đấy.

Biến sẽ giúp chúng ta lưu trữ và quản lý các giá trị trong chương trình.

Trong Java, Biến cũng là đại diện cho một đối tượng và đối tượng này phải được xác định là thuộc Kiểu dữ liệu nào. Sẽ giống với phương trình x kia, nhưng đề bài phải ghi rõ x là số nguyên, số thực hay số phức để người làm bài người ta còn biết.

Có các kiểu dữ liệu nguyên thuỷ (primitive) như sau:

* boolean: là kiểu logic, chỉ có 2 giá trị true hoặc false
* char: kiểu ký tự, chỉ chứa đc được một ký tự, được định nghĩa trong dấu ngoặc đơn '
* int : số nguyên (1,2,3, ..)
* long: số nguyên, lớn hơn int. (sẽ giải thích ở dưới)
* float: số thực (1.5, 2.5, ..).
* double: số thực, lớn hơn float.

Ngoài ra còn 2 kiểu dữ liệu nhỏ hơn int là byte và short. Thì mình sẽ nói sau hah. Còn trước mắt tập trung vào các nhóm chính kia đã.

Tiếp đến là kiểu dữ liệu cao cấp hơn gọi là Object mà đặc trưng nhất là String.

* String: Một chuỗi các ký tự, được định nghĩa trong dấu ngoặc kép "". vd String a = "Hellooo world~~~" (Nhớ tới ví dụ ở [Bài #1](https://loda.me/java-basic-1-gioi-thieu-java-cai-dat-moi-truong-va-hellooo-world-loda1553005203072) hem các bạn)

Mọi loại dữ liệu đều có một cái gọi là Giá trị mặc định, khi các bạn không cung cấp cho nó giá trị, nó sẽ tự có 1 giá trị mặc định.

| KIỂU DỮ LIỆU | GIÁ TRỊ MẶC ĐỊNH | KÍCH THƯỚC |
| --- | --- | --- |
| [boolean](https://loda.me/cef5bb47048c4572b3f1fa20fd104067) | false | 1 bit |
| [char](https://loda.me/7e7768f76355488fbe82d2070b4ebe46) | '\u0000' | 2 byte |
| [byte](https://loda.me/31579ab3d69d4f2faf215b80f5ad34b2) | 0 | 1 byte |
| [short](https://loda.me/4ccb3aa006c6437b98067d03c836b3eb) | 0 | 2 byte |
| [int](https://loda.me/abef18f1ae504de2a884f473647cdc15) | 0 | 4 byte |
| [long](https://loda.me/dcc537fe55f14c0597e6b1138745eaed) | 0L | 8 byte |
| [float](https://loda.me/eb6da65170a1446b93b3193d38401c07) | 0.0f | 4 byte |
| [double](https://loda.me/e43660e9a7e443cbb1b547d69f4611ca) | 0.0d | 8 byte |
| [String](https://loda.me/2e47444a0f354bee98f646320c715ec0) | null |  |

Ở đây bạn sẽ thấy có chữ L f d sau số 0. Đó là những ký tự đánh dấu cho Java phân biệt là số 0L là số 0 nhưng dạng long, f là dạng float, d là double.

**Cách khai báo**

Để khai báo biến, bắt buộc trước đó bạn phải chỉ cho nó kiểu dữ liệu mà nó sẽ nhận, ngoài ra có thể có giá trị hoặc không.

* Cách 1: [kiểu\_dữ\_liệu][tên\_biến];
* Cách 2: [kiểu\_dữ\_liệu][tên\_biến] = [giá\_trị];

int a, b, c; // Khai báo 3 biến có kiểu dữ liệu int

float b = 4.5f, c = 4f; // Khai báo 2 biến có kiểu dữ liệu float với giá trị ban đầu. ở đây biến `c` sẽ được hiểu là c = 4.0

double c = 4444.3;

char t = 'c';

String e = "Hello";

**Cách đặt tên**

Trong Java, tuy không bắt buộc, nhưng chúng ta luôn thống nhất với nhau cách đặt tên biến theo một nguyên tắc, để đảm bảo khi đọc code sẽ có tính nhất quán và chuẩn chỉnh.

* Tên biến phải tuân theo quy tắc lạc đà (Camel Case): đó là chữ cái đầu tiên của từ đầu tiên phải viết thường và chữ cái đầu tiên của các từ tiếp theo phải viết hoa, ví dụ: listStudent, minScore.

Chi tiết các bạn xem [ở đây](https://www.oracle.com/technetwork/java/codeconventions-135099.html) nhé, nói ra khá dài, nhưng nắm được cái ý ở trên của mình là cũng khá ổn r.

**Phạm vi sử dụng**

Một khi bạn đã khai báo biến, thì bạn có thể sử dụng nó trong những Phạm vi mà nó khả dụng. ?? 😀?? Cùng nhìn ví dụ ở dưới nhé.

Ví dụ:

public static void main(String[] args){

// khai bao so nguyen `a`

int a;

// Gán giá trị cho a, bạn sử dụng toán tử `=`

// Sử dụng biến a bình thường

a = 124214;

// lấy a và cộng thêm 1,, rồi gán ngược lại giá trị đó vào a :D

// Sử dụng biến a bình thường

a = a + 1;

}

// Gán lại giá trị cho a = 100 - 10;

// Chương trình lỗi

a = 100 - 10;

Phạm vi (Scope) là đây các bạn ạ, chính là 2 cái dấu {}, khi bạn khai báo một biến a trong 2 cái dấu { } thì bạn chỉ có thể sử dụng ở trong nó thôi, ra ngoài nó sẽ không hiểu a là thằng nào và từ đâu chui ra.

Biến không thể sử dụng ngoài, nhưng nó có thể được sử dụng ở bên trong những scope mà nó chứa hoặc cùng cấp với nó.

public class Calculation{

// Khai báo a ở ngoài main, cái `public static` là cần thiết nhé, còn chi tiết thì chúng ta sẽ học ở các bài sau.

public static int a = 5;

public static void main(String[] args){

// thay đổi a, ở trong, vẫn okie.

a = 10;

// Biến a có thể sử dụng trong các `scope` con của nó

// Làm gì biến a ở đây cũng được, biến đổi nó.

// gán giá trị biến a vào b;

int b = a;

System.out.println(b);

}

}

**Toán tử**

Khi đã xác định các Biến trong chương trình, bạn có thể sử dụng toán tử để thay đổi các giá trị. Các toán tử thì khá đơn giản, giống môn toán bình thường thôi. Với các kiểu nguyên thuỷ (primitive) ta có:

public class Calculation{

public static void main(String[] args){

int a;

int b = 5;

int c = a + b; // c = 0 + 5 cộng

int d = a - b; // d = 0 - 5 trừ

int f = a \* 5; // f = 0 x 5 nhân

int g = a / 5; // g = 0 : 5; chia

}

}

Còn với String thì bạn có thể sử dụng + để ghép 2 chuỗi mà thôi. Còn các toàn tử còn lại không được sử dụng với String

public class Calculation{

public static void main(String[] args){

String a = "Hello"

String b = "World"

// Mình đã nối 3 xâu là "Hello" + " " (Khoảng trắng) + "World" lại với nhau

System.out.println(a + " " + b);

String c = a + 5; // String cộng với một số nguyên?

System.out.println(c); // Kết quả sẽ là: "Hello 5" :V

// Bạn sẽ hiểu là khi cộng String với một số, số đó sẽ bị chuyển thành String và nối vào sau.

}

}

Ở trên có một ví dụ về việc cộng một String với int. Rất kì lạ phải không, 2 kiểu dữ liệu khác nhau khi tính toán với nhau thì sẽ được Java xử lý bằng cách Ép kiểu.

**Ép kiểu dữ liệu**

Nhìn vào ví dụ sau, bạn sẽ rõ.

public class Calculation{

public static void main(String[] args){

int a = 2;

float b = 3.5f; // dùng chữ f để nó hiểu đây là 3,5 float chứ k phải 3,5 double

float c = a + b; // c = 5.5

int d = a + b; // báo lỗi. Vì sao?

// vì java đang hiểu 2 + 3.5 nó sẽ ép thành 5.5 là float. Bây giờ gán nó vào số nguyên thì sẽ như này int = float?

// Để gán được bạn cần sử dụng ép kiểu

int d = (int) a + b; // d = 5

// a + b = 5.5 => ép thành (int) => 5 (lấy phần nguyên thôi)

char character = '5';

int number = (int) character; // number = 53. Why?

// Vì ép `char` thành `int` thì nó sẽ không chuyển chữ thành số, mà nó sẽ kiếm tra '5' là ký tự ASCII thứ bao nhiêu trong máy tính, và trả lại số thứ tự đó.

float = (float) 5; // => 5.0

}

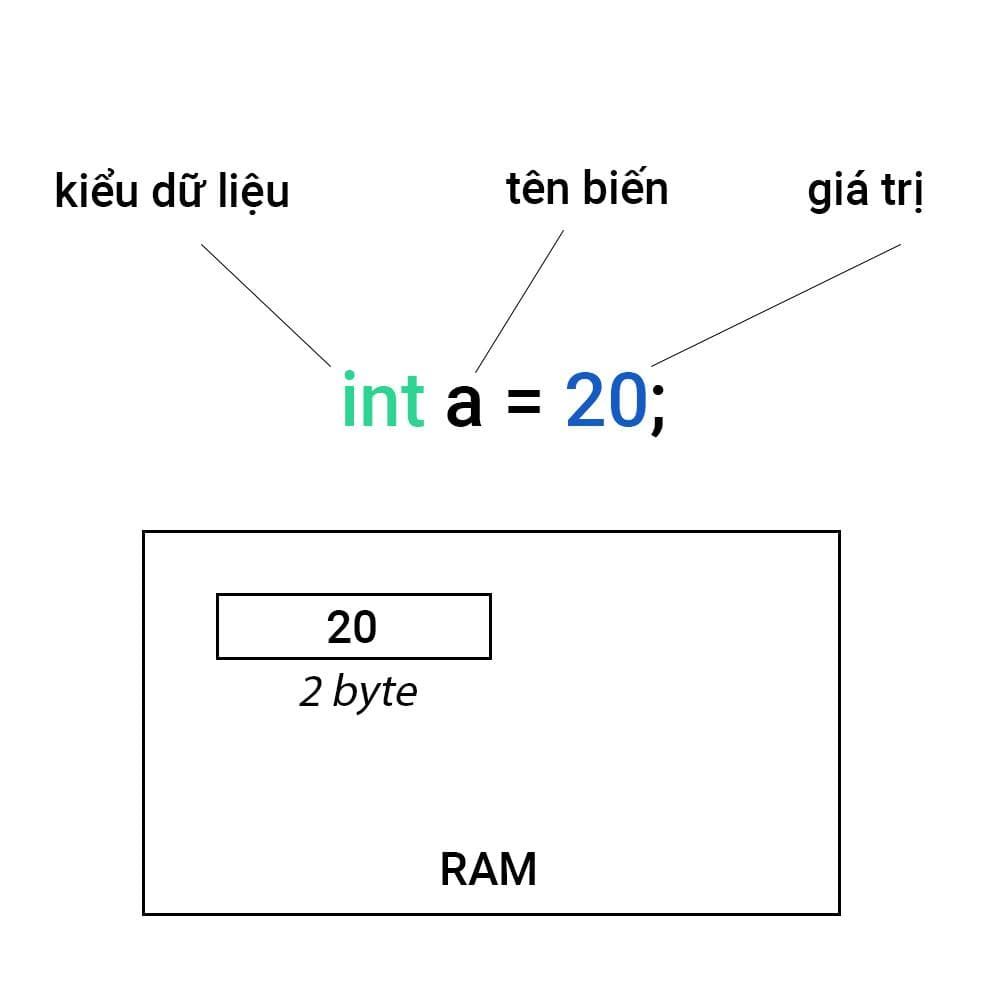
}

Tới đây là các bạn đã có thể sử dụng được Biến trong Java rồi đó, có thể sử dụng làm bài tập được rồi kakakakaka :D, còn phần dưới đây mình sẽ nói thêm về bản chất của Biến và giới hạn giá trị của nó.

**Bản chất của biến (Nói thêm)**

Khi các bạn khai báo một biến int trong chương trình của mình và sử dụng lung tung khắp mọi nơi, thì bạn có biết cái biến int ý ở đâu lòi ra không :))

Về bản chất, Biến sẽ là một vùng nhớ trong thiết bị vật lý mà dễ nhất là để trong ram. và khi bạn cho nó một giá trị, nó sẽ lưu trữ số đó vào ram, và cần thì lấy lên.



Vậy để ram biết bạn muốn lưu cái gì thì bạn phải khai báo cho nó. Ví dụ bạn bảo tôi cần một số nguyên int. Thì máy tính hiểu là mình cần lưu trữ một số nguyên bình thường, không quá lớn, nó sẽ cho bạn 4 byte trong Ram thích lưu gì thì lưu. nhưng không được vượt quá 4 byte.

4 byte = 32 bit, bỏ đi 1 bit đầu tiên để đánh dấu là số âm hay dương, thì còn 31 bit => số lớn nhất mà biến int lưu trữ được là 2^31 - 1 = 2147483647

Từ đây, bạn sẽ hiểu vì sao có số long, vì nhu cầu lưu số lớn hơn thì long được cấp tận 8 byte.

Còn trường hợp đặc biệt như String thì tuỳ giá trị của nó có bao nhiêu ký tự, mà Ram sẽ cấp tương ứng bấy nhiêu byte