

● JKad15D 「のび太 VS 出木杉③」 (のび太は JKad15A1 で実装)

2 つの整数を入力し商と余りを表示する処理を作成せよ。商と余りの情報は出木杉 (dekisugi メソッド) に渡して、出木杉が計算するものとする。

出木杉の仕様

書式	処理
public static void dekisugi(int n1, int n2)	$n1 \div n2$ を計算し、商と余りを表示する。

課題完成時の画面

1 より大きい整数 1 を入力してください>10
1 より大きい整数 2 を入力してください>3
10÷3 の商と余りを計算します！
出木杉くん、よろしく！
そんなの簡単さ！
商は 3、余りは 1 だよ。

● JKad15C 「1 から n まで足してみよう！④ (メソッド編)」

1 から入力された数値 n まで加算する処理を作成せよ。なお、加算する処理はメソッドにすること。

メソッドの仕様

書式	処理
public static int integrate(int n)	1 から n まで加算し、その結果を返す。

課題完成時の画面

1 より大きい整数を入力してください>10
1 から 10 まで加算します！
合計は 55 です！！

● JKad15B 「最大公約数と最小公倍数②」 (JKad12X と同じ)

最大公約数 (GCD) と最小公倍数 (LCM) を求めるメソッドを作成し、ふたつの整数の GCD と LCM を表示する処理を作成せよ。なお、課題フォルダに JKad12A1 (最大公約数) と JKad12A2 (最小公倍数) の解答例があるので最大公約数と最小公倍数を求める処理の参考にすること。

メソッドの仕様

書式	処理
public static int calcGCD(int n1, int n2)	n1 と n2 の最大公約数を返す
public static int calcLCM(int n1, int n2)	n1 と n2 の最小公倍数を返す

課題完成時の画面

ひとつめの整数を入力してください>**30**
ふたつめの整数を入力してください>**50**
最大公約数は 10 です！
最小公倍数は 150 です！

● JKad15A1 「のび太 VS 出木杉④」 (JKad15D のコピー OK)

JKad15D にのび太を追加し、解答者を出木杉とのび太から選択できるようにせよ。

のび太の仕様

書式	処理
public static void nobita(int n1, int n2)	$n1 \div n2$ を計算し、商と余りを表示する。ただし「/」と「%」は使えない。

課題完成時の画面

1 より大きい整数 1 を入力してください>**10**
1 より大きい整数 2 を入力してください>**3**
10÷3 の商と余りを計算します！
誰を呼び出しますか？ (1 : 出木杉、それ以外 : のび太) >**2**
あ、そうか！リンゴを食べる問題と一緒にだね！
商は 3、余りは 1 だよ。

● JKad15A2 「1 から n まで足してみよう！⑤（再帰呼び出し編）」（JKad15C のコピペ OK）

1 から入力された数値 n まで加算する処理を作成せよ。なお、加算処理のメソッドの仕様は以下の通り。

メソッドの仕様

書式	処理
public static int integrate(int n)	n が 1 のとき、1 を返す。 そうでないとき、n-1 を計算するメソッドを呼び出し、そのメソッドの戻り値に n を加算して返す。

課題完成時の画面（JKad15C と同じ）

● JKad15S 「出木杉 VS しずか②」

チーム数を入力するとトーナメントの試合数を教えてくれる処理を作成せよ。なお、解答者は出木杉としづかから選択できるものとする。

出木杉君としずかちゃんの仕様（具体的な処理は下記「出木杉君の考え」「しずかちゃんの考え」を参照）

書式	処理
public static void dekisugi(int n)	出木杉君の処理。チーム数nを渡すと試合数を計算して表示する。
public static void sizuka(int n)	しずかちゃんの処理。チーム数nを渡すと試合数を計算して表示する。

課題完成時の画面（出木杉君）

トーナメントのチーム数を入力してください>**13**

誰が試合数を計算しますか？（1：出木杉、それ以外：しずか）>**1**

そんなの簡単さ！

1 回戦の試合数：6	不戦勝：1	残りチーム：7
2 回戦の試合数：3	不戦勝：1	残りチーム：4
3 回戦の試合数：2	不戦勝：0	残りチーム：2
4 回戦の試合数：1	不戦勝：0	残りチーム：1

トータルの試合数は12だよ。

【出木杉君の考え】(13 チームの場合)

1 回戦は 12 チームで対戦して 1 チームが不戦勝。12 チームで対戦するので試合数は 6、残りチームは不戦勝チームと合わせて 7 チームになる。

2 回戦は 7 チームだから、6 チームで対戦して 1 チームが不戦勝。6 チームで対戦するので試合数は 6、残りチームは不戦勝のチームと合わせて 4 チームになる。

これを残りチームが 1 チームになるまで繰り返せばいいんだ。

課題完成時の画面（しずかちゃん、●の部分は考えること）

トーナメントのチーム数を入力してください>**13**
 誰が試合数を計算しますか？（1：出木杉、それ以外：しずか）>**2**
 そんなの簡単よ！
 1 試合すれば●●●●●●●●●●●●●●●●
 最後に1 チーム残るためには12 試合すればいいのよ。

【しずかちゃんの考え】 (13 チームの場合)

1 試合すれば●●●●●●●●●●だから、試合数は12試合だわ。
(各自で考えること)

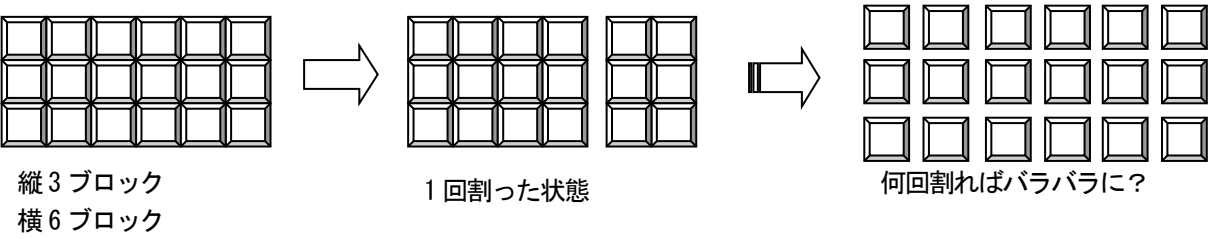
● JKad15X 「出木杉 VS しずか③」

板チョコを切れ目で割ってバラバラにしたい。何回割ればバラバラになるのか教えてくれる処理を作成せよ。なお、解答者は出木杉としずかから選択できるものとする。

出木杉君としずかちゃんの仕様（具体的な処理は下記「出木杉君の考え」「しずかちゃんの考え」を参照）

書式	処理
public static void dekisugi(int v, int h)	出木杉君の処理。縦のブロック数 v と横のブロック数 h を渡すと何回割ればいいのか計算して表示する。
public static void sizuka(int v, int h)	しずかちゃんの処理。縦のブロック数 v と横のブロック数 h を渡すと何回割ればいいのか計算して表示する。
public static int breakOff(int v, int h)	出木杉君が使ったメソッド。縦のブロック数 v と横のブロック数 h を渡すと何回割ればいいのか戻り値として返してくれる。

チョコレートの仕様（縦：3×横：6 の場合）



課題完成時の画面（出木杉君）

板チョコを割っていきます！
縦のブロックはいくつですか？>3
横のブロックはいくつですか？>6
誰が割る回数を計算しますか？（1：出木杉、それ以外：しずか）>1
これはなかなか手ごわいぞ！
とりあえず1回割って、それぞれをまた割って、全部で17回だ！

【出木杉君の考え】

縦と横のブロック数を渡すと何回割ればいいのかを教えてくれるメソッド（breakOff メソッド）を作ればいいんだ。breakOff メソッドの中で1回割ったと仮定して、割ったあとのチョコレートに対し、再び breakOff メソッドを呼び出せばいいんだ。

課題完成時の画面（しずかちゃん、●の部分は考えること）

板チョコを割っていきます！
縦のブロックはいくつですか？>3
横のブロックはいくつですか？>6
誰が割る回数を計算しますか？（1：出木杉、それ以外：しずか）>2
そんなの簡単よ！
1回割れば●●●●●●●●●●
全部で18個にするには17回割ればいいのかよ。

【しずかちゃんの考え】

1回割れば●●●●●●●●●●になるのだから、全部バラバラにするには●●回割ればいいのかだわ。