# **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÷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のコピペ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÷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 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）**

【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。

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。

**課題完成時の画面（しずかちゃん、●の部分は考えること）**

【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）

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。

# **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）**

何回割ればバラバラに？

1回割った状態

縦3ブロック

横6ブロック

**課題完成時の画面（出木杉君）**

【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。

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！

**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●●回割ればいいんだわ。