第十一回　演習問題（デバッグ等）

諸注意

- 「Kadai1.java」，「Monster.java」，「MonsterManager.java」，「Clicker.java」の４ファイルを Web から提出する．
- 指定した処理が正常に動くようであれば，適宜独自メソッド等を実装してくれて構わない．
- コピペ発覚時は見せた側も見せてもらった側も両方０点とする．
- 必ずコンパイルエラーのない状態で提出すること（自動採点したいのでコンパイルエラーがあると，全て０点になってしまう）．
- 課題の途中で提出することになった場合，コンパイルエラーさえ出なければ，課題の途中の状態で提出してくれて構わない．一部のメソッドだけが実現できていない場合，コンパイルエラー出ないならばそのままの状態で提出してくれてよい．
- 主にコンソール出力で評価しているため，デバッグに用いたようなコンソール出力が残っていないように気をつけること．基本的にコンソール出力を指定しない限りは，課題内でコンソール出力はないものとする．
- Package は使わないこと（デフォルトパッケージで実装する）．Package で実装すると，自動採点がうまくいきません．

課題 1

1	問題設定	引数の list から最大値を取得するメソッド getMax()を部下が開発したのだがどうも間違っている様子である．次の手順でデバッグせよ．getMax()は以下のコードを Kadai1.java に張り付けること．

	getMax()	`public static int getMax(int[] array){`
		`  int max = 0;`
		`  for(int val : array){`
		`  if(max > val) max = val;`
		`  }`
		`  return max;`
		`}`

	諸注意	● メソッドは static とすること． ● メソッドは全クラス，全パッケージからアクセスできるようにすること．
	手順１	何が誤っているのか次のvoid check()メソッドの実装を通して見極めよ．なお，半角スペース等は不要であり，[改行]は改行コードを意味する． 以下void check()の処理手順とする． 1．int型配列arrayを作成する． （初期値は，0,1,2,3,4,5,6,7,8,9とする） 2．次の書式でarrayの中身をコンソール出力する． 0,1,2,3,4,5,6,7,8,9,[改行] 3．getMax(array)をprintln()でコンソール出力する．
	手順２	手順１の結果を経て，デバッガの機能を用いて getMax()をデバッグせよ．デバッガの機能を使ったかどうかを我々が確認するすべはないが，是非練習だと思って 1．ブレークポイントを立てて 2．デバッグを実行し 3．ステップ・イン，ステップ・オーバーを活用し 4．バグの原因個所を見極めよ．
	手順３	引数によっては最大値が定義できない場合が存在する． 1．どのような引数のケースか考察せよ． 2．getMax()のはじめで該当の引数をif文で捉え，意図的にIllegalArgumentExceptionを発生させよ． エラーメッセージは「引数が異常です」とせよ．
	採点基準	0．check()が正しく実装されている． 1．バグ１が改善されている． 2．バグ２が改善されている． 3．バグ３が改善されている． ※バグは実は３つある．

課題２

2-1	問題設定	StringBuilder を用いた高速化を色々試してみよう.
	クラス名	Monster
	フィールド	name　：名前（文字列型） hp　：体力（int 型） ap　：攻撃力（int 型） mp　：魔法力（int 型） com　：コメント（文字列型）
	コンストラクタ	上記すべてのフィールドを上から順に初期化する.
	諸注意	● フィールドは全て隠蔽すること.
	メソッド	public String toStringPlus() テスト例のような文字列を返す. ただし，StringBuilderは使わずに＋で結合する. public String toStringBuilder() テスト例のような文字列を返す. ただし，文字列はStringBuilderを用いて結合する. ※テスト例の括弧は半角であり，括弧閉じの後にのみ半角スペースが1つ挿入されるものとする.
	テスト例	Main.javaのmain()メソッドにでも貼り付けて実行. Monster m = new Monster("はせがわ", 10, 50, 2, "期末テスト頑張って．"); System.out.println(m.toStringPlus()); System.out.println(m.toStringBuilder());
	テスト例出力	はせがわ[10,50,2] 期末テスト頑張って. はせがわ[10,50,2] 期末テスト頑張って.

<table>
<tr><td rowspan="6">2-2</td><td>問題設定</td><td>StringBuilder を用いた高速化を色々試してみよう.</td></tr>
<tr><td>クラス名</td><td>MonsterManager</td></tr>
<tr><td>static
フィールド</td><td>monsters　：モンスターの一覧を管理するリスト
（ArrayList<Monster>型）</td></tr>
<tr><td>諸注意</td><td>● フィールドは全て隠蔽すること.
● メソッドはどこからでもアクセスできること.
● フィールドもメソッドも全て static とする.</td></tr>
<tr><td>メソッド</td><td>void add(Monster m)
　引数のインスタンスをstaticフィールドに追加する.

String toStringPlusPlus()
　staticフィールドの中身全てを半角カンマ区切りで結合し戻り値として返す．ただし，各要素を文字列化するときにtoStringPlus()を用い，各要素間の結合にも＋を用いよ.

String toStringPlusBuilder()
　staticフィールドの中身全てを半角カンマ区切りで結合し戻り値として返す．ただし，各要素を文字列化するときにtoStringBuilder()を用い，各要素間の結合には＋を用いよ.

String toStringBuilderPlus()
　staticフィールドの中身全てを半角カンマ区切りで結合し戻り値として返す．ただし，各要素を文字列化するときにtoStringPlus()を用い，各要素間の結合にはStringBuilderを用いよ.

String toStringBuilderBuilder()
　staticフィールドの中身全てを半角カンマ区切りで結合し戻り値として返す．ただし，各要素を文字列化するときにtoStringBuilder()を用い，各要素間の結合にもStringBuilderを用いよ.</td></tr>
<tr><td>テスト例</td><td>Main.javaのmain()メソッドにでも貼り付けて実行.

MonsterManager.add(new Monster("は", 10, 50, 2, "期末"));
MonsterManager.add(new Monster("せ", 10, 50, 2, "ですと"));
MonsterManager.add(new Monster("が", 10, 50, 2, "がんば"));
MonsterManager.add(new Monster("わ", 10, 50, 2, "って"));

System.out.println(MonsterManager.toStringPlusPlus());
System.out.println(MonsterManager.toStringPlusBuilder());
System.out.println(MonsterManager.toStringBuilderPlus());
System.out.println(MonsterManager.toStringBuilderBuilder());</td></tr>
<tr><td>テスト例
出力</td><td>は[10,50,2]期末,せ[10,50,2]ですと,が[10,50,2]がんば,わ[10,50,2]って,
は[10,50,2]期末,せ[10,50,2]ですと,が[10,50,2]がんば,わ[10,50,2]って,
は[10,50,2]期末,せ[10,50,2]ですと,が[10,50,2]がんば,わ[10,50,2]って,
は[10,50,2]期末,せ[10,50,2]ですと,が[10,50,2]がんば,わ[10,50,2]って,</td></tr>
</table>

課題３（サービス問題：課題２までで既に１００点ある）

３	問題設定	下記の URL から働クリッカーというアプリを DL し使ってみよ．働クリッカーは５年ほど昔に一瞬流行った「クッキークリッカー」に類似するアプリである． 働クリッカーの特徴はここから先にある．同アプリでは JavaScript でチートコードを入力し，プログラムに働かせることができるのである．詳細は URL を参照してほしいが，基本動作は下記に示しておく．
	DL 先 URL	働クリッカー http://hsgw-nas.fuis.u-fukui.ac.jp/files/Hatara-Clicker-win64.zip 働クリッカー配布元（説明等） https://zeny.io/products/working-clicker/
	使用方法	【普通に遊ぶ】 ・「ゲームを始める」をクリックする ・「働く」をクリックするとお金が稼げる． ・資格等を購入すると時給が上がったりする． 【チートコード】 ・「C」のキーを押すとコンソールが開く ・コンソール上部にコードを記述する． ・Restart→Run でプログラムを実行できる． ・総資産１億で一応ゲームクリア． ・コンソール右上の？にリファレンスがある．
	諸注意	基本的に Java や C 言語と同じように記述できるが，変数宣言時には型名を全て var とする点だけ JavaScript は特殊である． なお，本日学習した高速化はあまり関係ない．ただ面白いからサービス問題として載せておくだけである．
	採点基準	ゲームクリアに 10 秒をきった場合，提出を受け付ける．該当のソースコードをコピペし，「Clicker.java」として提出せよ（Java のコードじゃないのでエラーが出るかもしれないが気にしなくて良い）． なお，長谷川が適当に遊んだ結果，長谷川の保有する PC のスペックで，0.409s を達成した．