# **J2Kad03D1「コンストラクタ」（入門編P.165「コンストラクタとは」）**

モンスターを表すMonsterクラスが準備されている。**リスト1**を入力して実行したところ、最初のshowDataメソッドの呼び出しで、名前が「null」、HPが「0」と表示される。名前とHPにデフォルト値（「まだないよ～」と「1」）が設定されるようにMonsterクラスにコンストラクタを追加せよ。

**追加するメソッド（Monsterクラス）**

|  |  |
| --- | --- |
| 書式 | 仕様 |
| public Monster() | コンストラクタ。setDataメソッドを使ってnameに「まだないよ～」、hpに1を設定する。 |

**リスト1：J2Kad03D1クラス**

public class J2Kad03D1 {

public static void main(String[] args) {

Monster m1 = new Monster();

m1.showData();

m1.setData("ピカチュウ", 20);

m1.showData();

}

}

**リスト1を入力して実行したときの画面 課題完成時の画面**

ぼくの名前はnull、HPは0だよ！

ぼくの名前はピカチュウ、HPは20だよ！

ぼくの名前はまだないよ～、HPは1だよ！

ぼくの名前はピカチュウ、HPは20だよ！

# **J2Kad03D2「引数付きコンストラクタ」（入門編P.168「コンストラクタのオーバーロード」）**

ヤドン（HP：30）を生成してデータ表示する処理を作成せよ。なおヤドンのデータ（名前とHP）の設定は、引数付きコンストラクタで行うこと。

**追加するメソッド（Monsterクラス）**

|  |  |
| --- | --- |
| 書式 | 仕様 |
| public Monster(String n, int h) | 引数付きコンストラクタ。setDataメソッドを使ってnameにn、hpにhを設定する。 |

**課題完成時の画面**

ぼくの名前はヤドン、HPは30だよ！

# **J2Kad03C「カプセル化」（入門編P.231「アクセス修飾子」、P.166「コンストラクタの例」）**

**リスト1**を入力して実行すると、ピカチュウの名前とHPが「ゲレゲレ」と「-1」になる。これではピカチュウがかわいそうだ！勝手にピカチュウの名前やHPを変更できないようにせよ。

**追加するメソッド（Monsterクラス）**

|  |  |
| --- | --- |
| 書式 | 仕様 |
| public void setName(String name) | nameのセッター。引数nameが「ゲレゲレ」のときは  「ゲレゲレなんていやだ～」と表示して、名前を設定しない。 |
| public void setHp(int hp) | hpのセッター。引数hpが0以下のときは「せめてHP、1はちょうだい！」と  表示して、1を設定する。 |

**リスト1：J2Kad03Cクラス**

public class J2Kad03C {

public static void main(String[] args) {

Monster m1 = new Monster("ピカチュウ", 20);

m1.showData();

m1.name = "ゲレゲレ";

m1.hp = -1;

m1.showData();

}

}

**リスト1まで入力したときの画面 課題完成時の画面**

ぼくの名前はピカチュウ、HPは20だよ！

ぼくの名前はゲレゲレ、HPは-1だよ！

ぼくの名前はピカチュウ、HPは20だよ！

ゲレゲレなんて名前はいやだ～

せめてHP、1はちょうだい！

ぼくの名前はピカチュウ、HPは1だよ！

# **J2Kad03B「アクセサ」（入門編P.72「3項演算子」）**

Profileクラスを作成し、ドラえもんとドラミちゃんのプロフィールを表示する処理を作成せよ。

**プロフィール**

|  |  |  |
| --- | --- | --- |
|  | 身長 | 体重 |
| ドラえもん | 129.3 | 129.3 |
| ドラミちゃん | 100.0 | 91.0 |

**仕様①**

J2Kad03BクラスのmainメソッドでドラえもんとドラミちゃんのProfileを引数付きコンストラクタで生成し、それぞれのフィールドを直接参照して表示する処理を作成せよ。

**Profileクラス（新規作成）の仕様①**

|  |  |  |
| --- | --- | --- |
|  | 書式 | 説明 |
| フィールド | public String name | 名前 |
| public double height | 身長 |
| public double weight | 体重 |
| メソッド | public Profile(  String name,  double height,  double weight  ) | コンストラクタ。  引数を対応するフィールドに設定する。 |

**仕様②**

Profileクラスのフィールドをすべてprivateにし、ゲッターを使って取得するように変更せよ。

**Profileクラスの仕様②**

|  |  |  |
| --- | --- | --- |
|  | 書式 | 説明 |
| メソッド | public String getName() | nameのゲッター。nameを返す。 |
| public double getHeight() | heightのゲッター。heightが120より小さいときは170を返す  （身長をごまかす）※3項演算子を使うこと |
| public double getWeight() | weightのゲッター。weightが100より大きいときは50を返す  （体重をごまかす）※3項演算子を使うこと |

**仕様①まで完成したときの画面 仕様②まで完成したときの画面**

ドラえもんのプロフィールです！

身長：129.3

体重：50.0

ドラミちゃんのプロフィールです！

身長：170.0

体重：91.0

ドラえもんのプロフィールです！

身長：129.3

体重：129.3

ドラミちゃんのプロフィールです！

身長：100.0

体重：91.0

Profileクラスからゲッターを使って取得した値を表示する。ドラえもんは体重を、ドラミちゃんは身長をごまかしている。

J2Kad03Bクラスから直接、身長や体重を参照するので

設定した値がそのまま表示される。

# **J2Kad03A「スタック！③」**

J2Kad03AクラスとStackクラスにJ2Kad02Xの完成版相当のプログラムが準備されている。ただしStackクラスのフィールドは他のクラスから直接アクセスできる状態になっている。Stackクラスのフィールドをprivateに変更し、プログラムが動作するように修正せよ。

**Stackクラスに追加するメソッド**

|  |  |
| --- | --- |
| 書式 | 説明 |
| public Stack(int size) | 要素数sizeの配列stackを生成し、スタックポインタ（sp）を0にする。 |
| public boolean isFull() | これ以上データを格納できないときtrue、格納できるときfalseを返す。 |
| public boolean isEmpty() | データがないときtrue、データがあるときfalseを返す。 |
| public int size() | スタックに格納されているデータ数を返す。 |
| public int get(int i) | スタックのｉ番目のデータを返す。 |

**課題完成時の画面（J2Kad02Xと同じ） （続き）**

スタック操作をします！

どうしますか？（1：push、2：pop、-1：終了）＞**1**

stack：4

どうしますか？（1：push、2：pop、-1：終了）＞**1**

stack：4 51

どうしますか？（1：push、2：pop、-1：終了）＞**1**

stack：4 51 26

どうしますか？（1：push、2：pop、-1：終了）＞**1**

stack：4 51 26 34

どうしますか？（1：push、2：pop、-1：終了）＞**1**

stack：4 51 26 34 71

どうしますか？（1：push、2：pop、-1：終了）＞**1**

stack：4 51 26 34 71 87

…

　（中略）

…

どうしますか？（1：push、2：pop、-1：終了）＞**1**

stack：4 51 26 34 71 87 18 5 17 94

どうしますか？（1：push、2：pop、-1：終了）＞**1**

スタックがいっぱいです！

stack：4 51 26 34 71 87 18 5 17 94

どうしますか？（1：push、2：pop、-1：終了）＞**2**

94を取り出しました！

stack：4 51 26 34 71 87 18 5 17

どうしますか？（1：push、2：pop、-1：終了）＞**2**

17を取り出しました！

stack：4 51 26 34 71 87 18 5

どうしますか？（1：push、2：pop、-1：終了）＞**2**

5を取り出しました！

stack：4 51 26 34 71 87 18

…

　（中略）

…

どうしますか？（1：push、2：pop、-1：終了）＞**2**

51を取り出しました！

stack：4

どうしますか？（1：push、2：pop、-1：終了）＞**2**

4を取り出しました！

stack：

どうしますか？（1：push、2：pop、-1：終了）＞**2**

データがありません！

stack：

どうしますか？（1：push、2：pop、-1：終了）＞**-1**

# **J2Kad03S「そうだ！ECC銀行へ行こう！！③」**

J2Kad03SクラスとAccountクラスにJ2Kad02Sの完成版相当のプログラムが準備されている。ただしAccountクラスのフィールド（口座名義・口座番号・預金残高・暗証番号）は他のクラス（ここではJ2Kad03Sクラス）から直接アクセスできる状態になっている（暗証番号もまる見えだ！）。Accountクラスのフィールドをprivateに変更し、プログラムが動作するように修正せよ。

**追加するメソッド（Accountクラス）**

|  |  |
| --- | --- |
| 書式 | 説明 |
| public Account(String name, int accountNumber,  int money, int secretNumber) | コンストラクタ。引数を対応するフィールドに設定する。 |
| public String getName() | 口座名義のゲッター。 |
| public int getAccountNumber() | 口座番号のゲッター。 |
| public int getMoney() | 預金残高のゲッター。 |
| public addMoney(int money) | 預金残高に引数moneyを加算する。 |
| public boolean checkSecretNumber(int secretNumber) | 暗証番号と引数secretNumberが一致していたらtrueを返す。  一致していなかったらfalseを返す。  ※暗証番号についてはゲッターを作らない。 |
| public void subMoney(int money) | 預金残高から引数moneyを減らす。 |

**課題完成時の画面（J2Kad02Sと同じ）**

そうだ！銀行へ行こう！！

誰が行きますか？（1：のび太、2：スネ夫、-1：誰もいかない）＞**2**

ECC銀行梅田本店へようこそ！

口座名義：スネ夫

口座番号：8901234

預金残高：10000000円

どうしますか？（1：預ける、2：引き出す、-1：帰る）＞**2**

暗証番号を入力してください＞**5678**

いくら引き出しますか？＞**5000000**

口座名義：スネ夫

口座番号：8901234

預金残高：5000000円

どうしますか？（1：預ける、2：引き出す、-1：帰る）＞**-1**

ありがとうございました！

誰が行きますか？（1：のび太、2：スネ夫、-1：誰もいかない）＞**-1**

# **J2Kad03X「そうだ！ECC銀行へ行こう！！④」※JKad03Sをコピーして作成**

信頼と実績のECC銀行に難波支店ができた！これで梅田本店と難波支店の2店舗体制になった。のび太とスネ夫が銀行へときにどちらへ行くのか選択できる処理を追加せよ。ECC銀行に関してはECCBankクラスを新規作成すること。

**ECCBankクラスの仕様（新規ファイル「ECCBank.java」）**

|  |  |  |
| --- | --- | --- |
|  | 書式 | 説明 |
| フィールド | private String name | 店名（初期値：設定しない） |
| メソッド | 引数付きコンストラクタ | 店名（梅田本店、難波支店）を設定する |
| public void gotoECCBank(Account account) | J2Kad03SのgotoBankメソッドに該当する処理 |
| … | （その他必要なメソッドあれば追加する） |

**課題完成時の画面**

そうだ！銀行へ行こう！！

誰が行きますか？（1：のび太、2：スネ夫、-1：誰もいかない）＞**1**

どの支店へ行きますか？（1：梅田本店、2：難波支店）＞**2**

ECC銀行難波支店へようこそ！

口座名義：のび太

口座番号：1234567

預金残高：0円

どうしますか？（1：預ける、2：引き出す、-1：帰る）＞**1**

いくら預けますか？＞**100**

口座名義：のび太

口座番号：1234567

預金残高：100円

どうしますか？（1：預ける、2：引き出す、-1：帰る）＞**-1**

ありがとうございました！

誰が行きますか？（1：のび太、2：スネ夫、-1：誰もいかない）＞**2**

どの支店へ行きますか？（1：梅田本店、2：難波支店）＞**1**

ECC銀行梅田本店へようこそ！

口座名義：スネ夫

口座番号：8901234

預金残高：10000000円

どうしますか？（1：預ける、2：引き出す、-1：帰る）＞**2**

暗証番号を入力してください＞**5678**

いくら引き出しますか？＞**1000000**

口座名義：スネ夫

口座番号：8901234

預金残高：9000000円

どうしますか？（1：預ける、2：引き出す、-1：帰る）＞**-1**

ありがとうございました！

誰が行きますか？（1：のび太、2：スネ夫、-1：誰もいかない）＞**-1**

店名を表示すること