**【練習問題】第７・８・９・１３章　アクセス修飾子④**

* **practice.practice\_print13.practice04パッケージ内に以下のクラスを作成しましょう。**

1. BankAccountクラスと、mainメソッドを持つMainクラスを作成してください。
2. BankAccountクラスに以下を記述してください。

・フィールド

|  |  |
| --- | --- |
| 型 | 変数名 |
| String | balance ※アクセス修飾子はpublic |

・コンストラクタ

　　　引数を1つ受け取り、フィールドにセット

1. Mainクラス内にmainメソッドを定義し、以下の処理を記述してください。

・BankAccountクラスのインスタンスを1つ生成し、「1000」をコンストラクタに渡す

・そのインスタンスのフィールドに「-5000」をセット

・実行結果例のように表示

|  |
| --- |
| 実行結果例 |
| 残高: -5000 |

**【練習問題】第７・８・９・１３章　アクセス修飾子⑤**

* **practice.practice\_print13.practice05パッケージ内に以下のクラスを作成しましょう。**

1. BankAccountクラスと、mainメソッドを持つMainクラスを作成してください。
2. BankAccountクラスに以下を記述してください。

・フィールド

|  |  |
| --- | --- |
| 型 | 変数名 |
| String | balance　※アクセス修飾子はpublic |

・コンストラクタ

　　　引数を1つ受け取り、フィールドにセット

・メソッド

　　　・getterであるgetBalance()メソッドを定義

1. Mainクラス内にmainメソッドを定義し、以下の処理を記述してください。

・BankAccountクラスのインスタンスを1つ生成し、「1000」をコンストラクタに渡す

・実行結果例のように表示

・「account.balance = -5000;」を記述するとエラーになることを確認

|  |
| --- |
| 実行結果例 |
| 残高: -5000 |

**【練習問題】第７・８・９・１３章　アクセス修飾子⑥**

* **practice.practice\_print13.practice06パッケージ内に以下のクラスを作成しましょう。**

1. BankAccountクラスと、mainメソッドを持つMainクラスを作成してください。

1. BankAccountクラスに以下を記述してください。

・フィールド

|  |  |
| --- | --- |
| 型 | 変数名 |
| String | balance　※アクセス修飾子はpublic |

・コンストラクタ

　　　引数を1つ受け取り、フィールドにセット

・メソッド

　　　・getterであるgetBalance()メソッドを定義

　　　・depositメソッド

　　　　　引数：int型1つ

　　　　　処理：引数の値が0より大きい場合はフィールドに加算、それ以外の場合は「throw new IllegalArgumentException("入金額は正の値にしてください");」を記述

1. Mainクラス内にmainメソッドを定義し、以下の処理を記述してください。

・BankAccountクラスのインスタンスを1つ生成し、「1000」をコンストラクタに渡す

・引数に「500」を渡し、depositメソッドを呼び出す

・実行結果例のように表示

・引数に「-200」を渡し、depositメソッドを呼び出す(例外を確認)

|  |
| --- |
| 実行結果例 |
| 残高: 1500 |

**【練習問題】第７・８・９・１３章　アクセス修飾子⑦**

* **practice.practice\_print13.practice06パッケージ内に以下のクラスを作成しましょう。**

1. BankAccountクラスと、mainメソッドを持つMainクラスを作成してください。

1. BankAccountクラスに以下を記述してください。

・フィールド

|  |  |
| --- | --- |
| 型 | 変数名 |
| String | balance　※アクセス修飾子はpublic |

・コンストラクタ

　　　引数を1つ受け取り、フィールドにセット

・メソッド

　　　・getterであるgetBalance()メソッドを定義

　　　・depositメソッドを定義

　　　　　引数：int型1つ

　　　　　処理：引数の値が0より大きい場合はフィールドに加算、それ以外の場合は「throw new IllegalArgumentException("入金額は正の値にしてください");」を記述

　　　・withdrawメソッドを定義

　　　　　引数：int型1つ

　　　　　処理：引数の値が0より大きく、残高が引数以上の場合、残高から引数の値を減算する。それ以外の場合は「throw new IllegalArgumentException("残高不足または不正な金額です");」を記述

1. Mainクラス内にmainメソッドを定義し、以下の処理を記述してください。

・BankAccountクラスのインスタンスを1つ生成し、「1000」をコンストラクタに渡す

・引数に「500」を渡し、depositメソッドを呼び出す

・実行結果例のように表示

・引数に「700」を渡し、withdrawメソッドを呼び出す

・引数に「1000」を渡し、withdrawメソッドを呼び出す(例外を確認)

|  |
| --- |
| 実行結果例 |
| 残高: 1500  残高: 800 |