

作成物の仕様

仕 様

1. Todoリストとは 忘れたくないタスクを登録しておくためのアプリです。実装する機能は、タスクの一覧表示、タスクの登録、タスクの削除です。
2. タスクの一覧表示 現在登録されているタスクの一覧を表示します。表示内容はタスクのIDとタスクのテキストです。
3. タスクの登録 タスクを1件登録します。登録されたタスクは一覧表示で表示されるようになります。現在登録されているタスクのうち、最も大きいIDに1を足した数をIDとして割り当てます。
4. タスクの削除 指定されたタスクを1件削除します。削除されたタスクは、一覧表示に表示されなくなります。
5. コマンドについて ユーザーは、コマンドを入力してアプリを操作します。ユーザーの入力を待ち受けする際は画面に「>」と表示し、ユーザーは使いたい機能に応じたコマンドを入力します。コマンドは下記表のとおりです。
6. 入力に関する制限 表に示したコマンド以外の入力があった場合は、「不正な入力です。」と表示して、入力待ち受けに戻ってください。addコマンドで追加できるタスクは20文字以内とします。

機 能

コマンド名	機能	書式	備考
show	タスクの一覧表示	show	-
add	タスクの登録	add 文字列	文字率はユーザーが登録したいタスクを示す。addと 文字列の間には半角スペースをいれる。
delete	タスクの削除	delete タスクのID	タスクのIDは、削除したいタスクのIDとする。deleteとタスクのIDの間には半角スペースをいれる。

動 作 例

```
>show
id      タスク
1       仕事する
2       買い物する

>add 掃除する
>show
id      タスク
1       仕事する
2       買い物する
3       掃除する

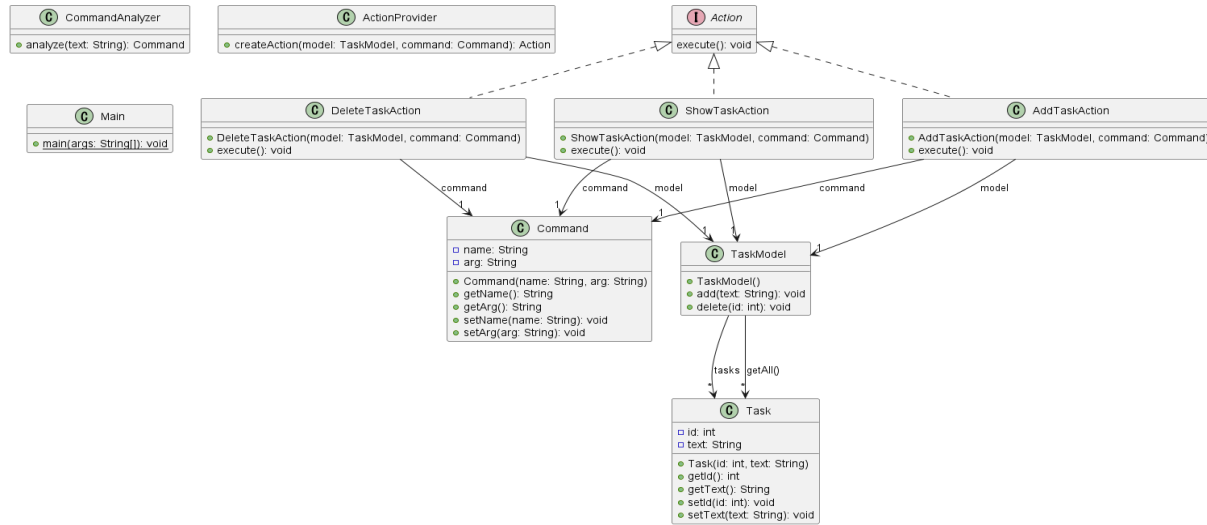
>delete 2
>show
id      タスク
1       掃除する
3       掃除する
```

作成日	23.01.12/井澤明美/1.0
-----	-------------------

修正日	23.01.21/井澤明美/1.1
-----	-------------------

No.	ファイル名	コード数	作成者	作成日	バージョン	修正者	修正日	バージョン
1	Action.java	1	井澤明美	22.01.12	1.0	テスト太郎	23.01.21	1.1
2	ActionProvider.java	1	井澤明美	22.01.13	1.0	テスト太郎	23.01.21	1.1
3	AddTaskAction.java	2	井澤明美	22.01.14	1.0	テスト太郎	23.01.21	1.1
4	Command.java	4	井澤明美	22.01.15	1.0	テスト太郎	23.01.21	1.1
5	CommandAnalyzer.java	1	井澤明美	22.01.16	1.0	テスト太郎	23.01.21	1.1
6	DeleteTaskAction.java	1	井澤明美	22.01.17	1.0	テスト太郎	23.01.21	1.1
7	Main.java	1	井澤明美	22.01.18	1.0	テスト太郎	23.01.21	1.1
8	ShowTaskAction.java	1	井澤明美	22.01.19	1.0	テスト太郎	23.01.21	1.1
9	Task.java	4	井澤明美	22.01.20	1.0	テスト太郎	23.01.21	1.1
10	TaskModel.java	3	井澤明美	22.01.21	1.0	テスト太郎	23.01.21	1.1
	合計	19						

全体のクラス図



ActionProvider

ActionProvider.java

クラス名

ActionProvider

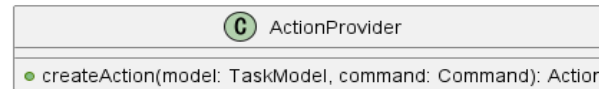
ファイル名

ActionProvider.java

説明

Commandオブジェクトに対し適切なActoinを選定する

クラス図



AddTaskAction

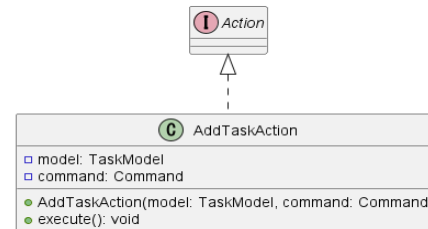
AddTaskAction.java

クラス名	AddTaskAction
------	---------------

ファイル名	AddTaskAction.java
-------	--------------------

説明	タスクを追加する
----	----------

クラス図



Command

Command. java

クラス名

Command

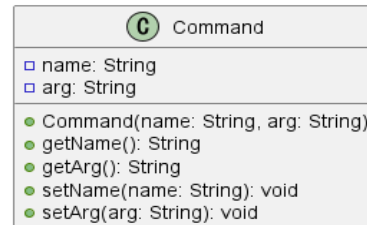
ファイル名

Command. java

説明

CommandAnalyzerクラスでチェックして文字列をコマンドオブジェクトに変換する

クラス図



CommandAnalyzer

CommandAnalyzer.java

クラス名

CommandAnalyzer

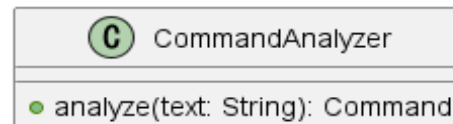
ファイル名

CommandAnalyzer.java

説明

ユーザーが入力したコマンドをチェックする

クラス図



DeleteTaskAction

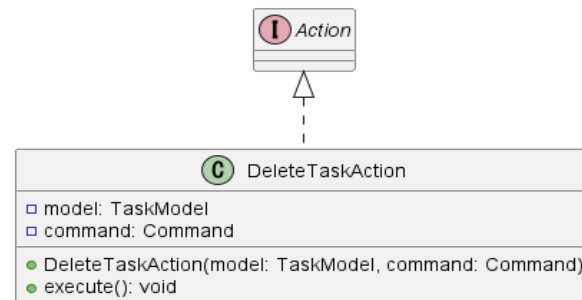
DeleteTaskAction.java

クラス名	DeleteTaskAction
------	------------------

ファイル名	DeleteTaskAction.java
-------	-----------------------

説明	登録したタスクを削除する
----	--------------

クラス図



ShowTaskAction

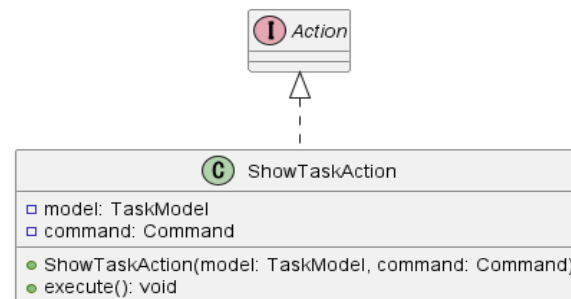
ShowTaskAction.java

クラス名	ShowTaskAction
------	----------------

ファイル名	ShowTaskAction.java
-------	---------------------

説明	登録したタスク一覧を表示させる
----	-----------------

クラス図



Task

Task.java

クラス名

Task

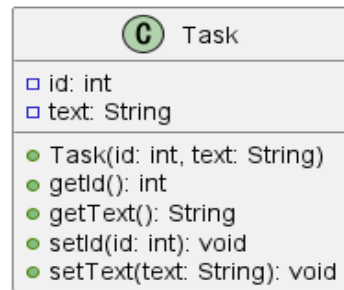
ファイル名

Task.java

説明

タスクを格納する

クラス図



TaskModel

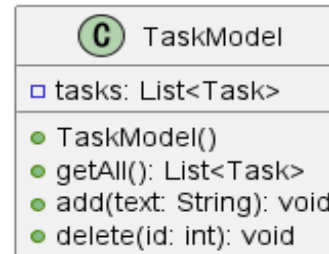
TaskModel.java

クラス名	TaskModel
------	-----------

ファイル名	TaskModel
-------	-----------

説明	各タスクの具体的な処理を記載する
----	------------------

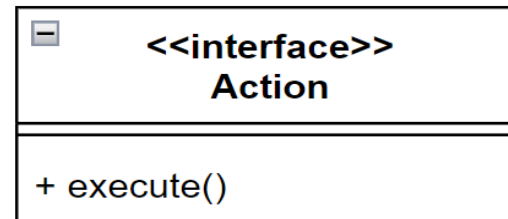
クラス図



Action

Actoin.java

クラス図



createAction

ActionProvider.java

構 文

```
public Action createAction(TaskModel model, Command command)
```

機 能

アクション振り分け

機 能 説 明

コマンドの内容に対して適切なアクションを振り分ける

戻 り 値

```
new ShowTaskAction(model, command)
new AddTaskAction(model, command)
new DeleteTaskAction(model, command)
null
```

備 考

処 理 説 明

- ・ コマンドの入力値がshowと等しい場合には、戻り値としてShowTaskActionインスタンスを返す
- ・ コマンドの入力値がAddと等しい場合には、戻り値としてAddTaskActionインスタンスを返す
- ・ コマンドの入力値がdeleteと等しい場合には、戻り値としてDeleteTaskActionインスタンスを返す
- ・ 該当するものがない場合には、nullを返す

ファイル名

ActionProvider.java

作 成 日

23.01.12/井澤 明美/1.0

修 正 日

23.01.12/井澤 明美/1.1

excute

AddTaskAction. java

構 文

```
public void execute() {  
    String argText -Commandクラスのargの内容を格納する  
}
```

機 能

追加処理の実行

機 能 説 明

タスクの追加処理を実行する

戻 り 値

なし

備 考

処 理 説 明

- ・ コマンドオブジェクトのarg内容を取得する
- ・ もし、argの文字の長さが20より大きい場合
- ・ 不正な入力であるとコンソールに出力する
- ・ それ以外の場合は、TaskModelのaddメソッドを使用しタスクを追加する

ファイル名

AddTaskAction. java

作 成 日

23.01.12/井澤 明美/1.0

修 正 日

23.01.12/井澤 明美/1.1

getName

Command.java

構文

```
public String getName()
```

機能

フィールドの値を取り出す

機能説明

フィールドの値「name」を取り出す

戻り値

this.name

備考

処理説明

- フィールドの値nameを戻り値として返す

ファイル名

command.java

作成日

23.01.12/井澤 明美/1.0

修正日

23.01.12/井澤 明美/1.1

getArg

Command. java

構 文

`public String getArg()`

機 能

フィールドの値を取り出す

機 能 説 明

フィールドの値「arg」を取り出す

戻 り 値

this.arg

備 考

処 理 説 明

・フィールドの値argを戻り値として返す

ファイル名

Command. java

作 成 日

23. 01. 12/井澤 明美/1. 0

修 正 日

23. 01. 12/井澤 明美/1. 1

setName

Command. java

構 文

```
public void setName(String name)
```

機 能

フィールドに値を代入する

機 能 説 明

フィールド(name)にパラメーターの値を代入する

戻 り 値

なし

備 考

処 理 説 明

- ・パラメーターの内容をフィールド変数に代入する

ファイル名

Command. java

作 成 日

23. 01. 12/井澤 明美/1. 0

修 正 日

23. 01. 12/井澤 明美/1. 1

setArg

Command. java

構 文

```
public void setArg(String arg)
```

機 能

フィールドに値を代入する

機 能 説 明

フィールド(arg)にパラメーターの値を代入する

戻 り 値

備 考

処 理 説 明

- ・パラメーターの内容をフィールド変数に代入する

ファイル名

Command. java

作 成 日

23.01.12/井澤 明美/1.0

修 正 日

23.01.12/井澤 明美/1.1

analyze

CommandAnalyzer.java

構 文

```
public Command analyze(String text) {  
    String[] splittedText    ←text.split(" ",2)で分割した内容を格納  
  
}
```

機 能

ユーザーのコマンド入力进行检查する

機能説明

ユーザーのコマンド入力、配列の長さに応じて检查する

戻 り 値

```
new Command(splittedText[0], splittedText[1]);  
new Command(splittedText[0], "")
```

備 考

処理説明

- ・splitメソッドを使用し、textを、区切り文字を半角スペース
- ・分割数を、2にした内容を配列変数splittedTextに格納する
- ・もし、配列の長さが1より大きい場合
- ・戻り値としてnew Command(splittedText[0], splittedText[1])を返す
- ・それ以外の場合は、
- ・戻り値としてnew Command(splittedText[0], "")を返す

ファイル名

CommandAnalyzer.java

作 成 日

23.01.12/井澤 明美/1.0

修 正 日

23.01.12/井澤 明美/1.1

excute

DeleteTaskAction.java

構 文

```
public void execute() {  
    int argId -Commandのargsの内容を格納する  
  
}
```

機 能

削除処理の実行

機 能 説 明

タスクの削除処理を実行する

戻 り 値

なし

備 考

処 理 説 明

- ・ try文
 - ・ argIdにコマンドクラスのargの内容をInteger.parseIntを使用し
 - ・ 文字列を数字に変換する
 - ・ TaskModelのdeleteメソッドを使用し、削除を実行する
- ・ catch(NumberFormatException e)
 - ・ コンソールに「不正な入力です」と出力する

ファイル名

DeleteTaskAction.java

作 成 日

23.01.12/井澤 明美/1.0

修 正 日

23.01.12/井澤 明美/1.1

excute

ShowTaskAction. java

構 文

```
public void execute() {  
    List<Task> tasks =TaskModelのallメソッドの内容を格納する  
}
```

機 能

一覧表示実行

機 能 説 明

タスクの一覧表示を実行する

戻 り 値

なし

備 考

処 理 説 明

- ・List<Task> tasks =TaskModelのallメソッドの内容を格納する
- ・コンソールに「id タスク」を出力する
- ・tasksに格納されている内容を拡張for文（for-each）で出力する
 - ・コンソールに「タスクのId + " " + タスク内容」を出力する
- ・タスクのId及びタスク内容は、getメソッドを使用して記載すること

ファイル名

ShowTaskAction. java

作 成 日

23. 01. 12/井澤 明美/1. 0

修 正 日

23. 01. 12/井澤 明美/1. 1

getId

Task.java

構 文

```
public int getId()
```

機 能

フィールドの値を取り出す

機 能 説 明

フィールドの値「id」を取り出す

戻 り 値

this.id

備 考**処 理 説 明**

・フィールドの値idを戻り値として返す

ファイル名

Task.java

作 成 日

23.01.12/井澤 明美/1.0

修 正 日23.01.12/井澤 明美/1.1

getText

Task.java

構 文

```
public String getText()
```

機 能

フィールドの値を取り出す

機 能 説 明

フィールドの値「text」を取り出す

戻 り 値

this.text

備 考**処 理 説 明**

・フィールドの値textを戻り値として返す

ファイル名

Task.java

作 成 日

23.01.12/井澤 明美/1.0

修 正 日

23.01.12/井澤 明美/1.1

setId

Task.java

構 文

```
public void setId(int id)
```

機 能

フィールドに値を代入する

機 能 説 明

フィールド(id)にパラメーターの値を代入する

戻 り 値

なし

備 考**処 理 説 明**

・パラメーターの内容をフィールド変数に代入する

ファイル名

Task.java

作 成 日

23. 01. 12/井澤 明美/1. 0

修 正 日23. 01. 12/井澤 明美/1. 1

setText

Task. java

構 文

```
public void setText(String text)
```

機 能

フィールドに値を代入する

機 能 説 明

フィールド(id)にパラメーターの値を代入する

戻 り 値

なし

備 考**処 理 説 明**

- パラメーターの内容をフィールド変数に代入する

ファイル名

Task. java

作 成 日

23. 01. 12/井澤 明美/1. 0

修 正 日

23. 01. 12/井澤 明美/1. 1

getAll

TaskModel.java

構 文

```
public List<Task> getAll()
```

機 能

一覧機能の処理

機 能 説 明

一覧表示をするための機能を実装する

戻 り 値

this.tasks

備 考

処 理 説 明

・フィールドの値tasksを戻り値として返す

ファイル名

TaskModel.java

作 成 日

23.01.12/井澤 明美/1.0

修 正 日

23.01.12/井澤 明美/1.1

add

TaskModel.java

構 文

```
public void add(String text) {  
    int currentMaxId =すでに登録されているタスクの最大id  
}
```

機 能

タスク追加の処理

機 能 説 明

追加処理を行うための機能を実装する

戻 り 値

なし

備 考

処 理 説 明

- ・ int currentMaxIdの初期値を0とする
- ・ 拡張for文 (for-each) を使用以下処理を実行する
 - ・ もし、取得したidがcurrentMaxIdより大きい場合
 - ・ currentMaxIdに取得したidを代入する
- ・ List<TaskModel>tasksにadd(List) を使用して、currentMaxIdに1を足した
- ・ ものを新規idとし、Taskインスタンス（新規Idとタスク内容）を格納する

ファイル名

TaskModel.java

作 成 日

23.01.12/井澤 明美/1.0

修 正 日

23.01.12/井澤 明美/1.1

delete

TaskModel.java

構 文

```
public void delete(int id)
```

機 能

削除の処理

機 能 説 明

削除処理を行うための機能を実装する

戻 り 値

なし

備 考

処 理 説 明

- ・ 拡張for文 (for-each) を使用以下の処理を実行する
- ・ もし、取得したidがパラメーターのidと同じ値であった場合
 - ・ Listのremoveメソッドを使用
 - ・ removeメソッド内ではindexOf()を使用し、削除対象を指定する
- ・ 戻り値を指定しないでreturn文を実行

ファイル名

TaskModel.java

作 成 日

23. 01. 12/井澤 明美/1. 0

修 正 日

23. 01. 12/井澤 明美/1. 1

main

Main. java

構 文

```
public static void main(String[] args)
```

機 能

実行

機 能 説 明

実行処理を行う

戻 り 値

なし

備 考**処 理 説 明**

- ・CommandAnalyzer、ActionProvider、TaskModelをインスタンス化する
- ・while(true)を使用し、処理を繰り返す
 - ・コンソールに「>」を出力する
 - ・String input変数を作成し、初期値にnullを代入
 - ・try文
 - ・BufferedReader inputuser = new BufferedReader(
new InputStreamReader(System.in) を使用しキーボード入力を受け付けるための処理を行う
 - ・inputにキーボードから入力された1行分の文字列をString型オブジェクトとして返す
readLineメソッドを使用し、入力内容をinputに格納する
 - ・catch (IOException e)
 - ・不正な入力処理とコンソールに出力後、continueする。
 - ・Command commandを宣言し、CommandAnalyzerのanalyzeメソッドの引数にinputを指定して初期化
- ・ActionProvider actionを宣言し、createActionメソッドの引数にインスタンス化したTaskModelの変数とcommandを指定する。
- ・もしactionがnullだった場合、コンソールに「不正な入力」と出力し、continueで処理を閉じる
- ・action変数からexecuteメソッドを使用する

ファイル名

Main. java

作 成 日

23. 01. 12/井澤 明美/1. 0

修 正 日

23. 01. 12/井澤 明美/1. 1

