

All Classes

Action

Controller

Dose

IAction

IDose

IMedicine

Medicine

PACKAGE

CLASS

TREE

DEPRECATED

INDEX

HELP

PREV CLASS

NEXT CLASS

FRAMES

NO FRAMES

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

Class Action

java.lang.Object

Action

All Implemented Interfaces:

IAction

```
public class Action
extends java.lang.Object
implements IAction
```

Field Summary

Fields

Modifier and Type	Field and Description
private IMedicine	medicine

Constructor Summary

Constructors

Constructor and Description
Action()

Method Summary

All Methods

Instance Methods

Concrete Methods

Modifier and Type	Method and Description
void	addDose (IDose dose) Adds a new dose.
java.util.ArrayList<java.lang.Double>	getCurrentConcentration (java.time.LocalDateTime time) Calculates concentration amount of doses at a specific time.
IMedicine	getMedicine () Retrieves the Medicine instance.
java.util.ArrayList<java.time.LocalDateTime>	getPeakConcentrationTime (java.lang.Boolean includeTestDoses) Time when the concentration is at its peak.
java.util.ArrayList<java.time.LocalDateTime>	getWhenToDose (java.lang.Double concentrationDesired) Determines time when the patient should take next dose.
void	loadFile (java.lang.String filename) Load a saved Medicine file.
void	newFile (java.lang.String name, java.time.LocalDateTime tMax, java.time.LocalDateTime halfLife) Create a new Medicine instance.
void	printCurrentConcentration (java.time.LocalDateTime time) Prints concentration amount of doses at a specific time.
void	printDoses () Prints all doses from Medicine's Dose array.
void	printMedicine () Print the name, tmax, and halfLife of the medicine.
void	printPeakConcentrationTime (java.lang.Boolean includeTestDoses) Print concentration peaks.

void	<code>printWhenToDose(java.lang.Double amountDesired)</code> Prints when the patient should take next dose.
void	<code>removeAllDoses()</code> Delete all doses from the dosages array.
void	<code>removeDose(int index)</code> Removes a dose from the dose from the doses array.
void	<code>removeTestDoses()</code> Removes all test doses.
void	<code>saveFile(java.lang.String filename)</code> Save Medicine to a file.

Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Field Detail

medicine

```
private IMedicine medicine
```

Constructor Detail

Action

```
public Action()
```

Method Detail

printDoses

```
public void printDoses()
```

Description copied from interface: `IAction`

Prints all doses from Medicine's Dose array.

Specified by:

`printDoses` in interface `IAction`

addDose

```
public void addDose(IDose dose)
```

Description copied from interface: `IAction`

Adds a new dose. Added doses can be type Dose (actual dose) or TestDose (what-if dose)

Specified by:

`addDose` in interface `IAction`

Parameters:

dose - Dose object.

removeDose

```
public void removeDose(int index)
```

Description copied from interface: `IAction`

Removes a dose from the dose from the doses array.

Specified by:

`removeDose` in interface `IAction`

Parameters:

index - Index of the array to be erased

removeAllDoses

```
public void removeAllDoses()
```

Description copied from interface: [IAction](#)

Delete all doses from the dosages array.

Specified by:

`removeAllDoses` in interface [IAction](#)

getCurrentConcentration

```
public java.util.ArrayList<java.lang.Double> getCurrentConcentration(java.time.LocalDateTime time)
```

Description copied from interface: [IAction](#)

Calculates concentration amount of doses at a specific time.

Specified by:

`getCurrentConcentration` in interface [IAction](#)

Parameters:

time - Specific time used to determine concentration amount.

Returns:

ArrayList with the concentration amounts.

printCurrentConcentration

```
public void printCurrentConcentration(java.time.LocalDateTime time)
```

Description copied from interface: [IAction](#)

Prints concentration amount of doses at a specific time.

Specified by:

`printCurrentConcentration` in interface [IAction](#)

Parameters:

time - Specific time used to determine concentration amount.

getPeakConcentrationTime

```
public java.util.ArrayList<java.time.LocalDateTime> getPeakConcentrationTime(java.lang.Boolean includeTestDoses)
```

Description copied from interface: [IAction](#)

Time when the concentration is at its peak. It allows the user to choose if test doses will be displayed or not.

Specified by:

`getPeakConcentrationTime` in interface [IAction](#)

Parameters:

includeTestDoses - Determines whether test doses will be displayed.

Returns:

ArrayList containing peak concentration peak values of each dose.

printPeakConcentrationTime

```
public void printPeakConcentrationTime(java.lang.Boolean includeTestDoses)
```

Description copied from interface: [IAction](#)

Print concentration peaks. Includes dose information and its concentration peak.

Specified by:

`printPeakConcentrationTime` in interface [IAction](#)

Parameters:

includeTestDoses - Determines whether test doses will be displayed.

getWhenToDose

```
public java.util.ArrayList<java.time.LocalDateTime> getWhenToDose(java.lang.Double concentrationDesired)
```

Description copied from interface: [IAction](#)

Determines time when the patient should take next dose.

Specified by:

`getWhenToDose` in interface [IAction](#)

Parameters:

concentrationDesired - Dose amount desired sought.

Returns:

Time when the patient have to take next dose.

printWhenToDose

```
public void printWhenToDose(java.lang.Double amountDesired)
```

Description copied from interface: [IAction](#)

Prints when the patient should take next dose. Includes dose information and time to take next dose.

Specified by:

`printWhenToDose` in interface [IAction](#)

Parameters:

amountDesired - Dose amount desired sought.

saveFile

```
public void saveFile(java.lang.String filename)
```

Description copied from interface: [IAction](#)

Save Medicine to a file. By default, save directory is same directory as the application. Dose and Medicine MUST implement Serializable for this feature to work.

Specified by:

`saveFile` in interface [IAction](#)

Parameters:

filename - Name of save file to be saved.

loadFile

```
public void loadFile(java.lang.String filename)
```

Description copied from interface: [IAction](#)

Load a saved Medicine file. By default, load directory is same directory as the application. Dose and Medicine MUST implement Serializable for this feature to work.

Specified by:

`loadFile` in interface [IAction](#)

Parameters:

filename - Name of the file to be loaded

newFile

```
public void newFile(java.lang.String name,  
                   java.time.LocalDateTime tMax,  
                   java.time.LocalDateTime halfLife)
```

Description copied from interface: [IAction](#)

Create a new Medicine instance.

Specified by:

`newFile` in interface [IAction](#)

Parameters:

name - Name of medicine

tMax - TMax of medicine.

halfLife - Half life of medicine

printMedicine

public void printMedicine()

Description copied from interface: [IAction](#)
Print the name, tmax, and halfLife of the medicine.

Specified by:
[printMedicine](#) in interface [IAction](#)

removeTestDoses

public void removeTestDoses()

Description copied from interface: [IAction](#)
Removes all test doses.

Specified by:
[removeTestDoses](#) in interface [IAction](#)

getMedicine

public [IMedicine](#) getMedicine()

Description copied from interface: [IAction](#)
Retrieves the Medicine instance.

Specified by:
[getMedicine](#) in interface [IAction](#)

Returns:
Medicine instance

[PACKAGE](#) **CLASS** [TREE](#) [DEPRECATED](#) [INDEX](#) [HELP](#)

[PREV CLASS](#) [NEXT CLASS](#) [FRAMES](#) [NO FRAMES](#)

[SUMMARY: NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#) [DETAIL: FIELD](#) | [CONSTR](#) | [METHOD](#)

Class Action

java.lang.Object
Action

All Implemented Interfaces:

IAction

```
public class Action
extends java.lang.Object
implements IAction
```

Field Summary

Fields	
Modifier and Type	Field and Description
private IMedicine	medicine

Constructor Summary

Constructors	
Constructor and Description	
Action()	

Method Summary

All Methods	Instance Methods	Concrete Methods
Modifier and Type		Method and Description
void		addDose(IDose dose) Adds a new dose.
java.util.ArrayList<java.lang.Double>		getCurrentConcentration(java.time.LocalTime time) Calculates concentration amount of doses at a specific time.
IMedicine		getMedicine() Retrieves the Medicine instance.
java.util.ArrayList<java.time.LocalTime>		getPeakConcentrationTime(java.lang.Boolean includeTestDoses) Time when the concentration is at its peak.
java.util.ArrayList<java.time.LocalTime>		getWhenToDose(java.lang.Double concentrationDesired) Determines time when the patient should take next dose.
void		loadFile(java.lang.String filename) Load a saved Medicine file.
void		newFile(java.lang.String name, java.time.LocalTime tMax,

```
java.time.LocalDateTime halfLife)
```

Create a new Medicine instance.

void	<code>printCurrentConcentration</code> (java.time.LocalDateTime time) Prints concentration amount of doses at a specific time.
void	<code>printDoses</code> () Prints all doses from Medicine's Dose array.
void	<code>printMedicine</code> () Print the name, tmax, and halfLife of the medicine.
void	<code>printPeakConcentrationTime</code> (java.lang.Boolean includeTestDoses) Print concentration peaks.
void	<code>printWhenToDose</code> (java.lang.Double amountDesired) Prints when the patient should take next dose.
void	<code>removeAllDoses</code> () Delete all doses from the dosages array.
void	<code>removeDose</code> (int index) Removes a dose from the dose from the doses array.
void	<code>removeTestDoses</code> () Removes all test doses.
void	<code>saveFile</code> (java.lang.String filename) Save Medicine to a file.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

medicine

```
private IMedicine medicine
```

Constructor Detail

Action

```
public Action()
```

Method Detail

printDoses

```
public void printDoses()
```

Description copied from interface: `IAction`

Prints all doses from Medicine's Dose array.

Specified by:

`printDoses` in interface `IAction`

addDose

```
public void addDose(IDose dose)
```

Description copied from interface: `IAction`

Adds a new dose. Added doses can be type `Dose` (actual dose) or `TestDose` (what-if dose)

Specified by:

`addDose` in interface `IAction`

Parameters:

`dose` - `Dose` object.

removeDose

```
public void removeDose(int index)
```

Description copied from interface: `IAction`

Removes a dose from the dose from the doses array.

Specified by:

`removeDose` in interface `IAction`

Parameters:

`index` - Index of the array to be erased

removeAllDoses

```
public void removeAllDoses()
```

Description copied from interface: `IAction`

Delete all doses from the dosages array.

Specified by:

`removeAllDoses` in interface `IAction`

getCurrentConcentration

```
public java.util.ArrayList<java.lang.Double> getCurrentConcentration(java.time.LocalTime time)
```

Description copied from interface: `IAction`

Calculates concentration amount of doses at a specific time.

Specified by:

`getCurrentConcentration` in interface `IAction`

Parameters:

`time` - Specific time used to determine concentration amount.

Returns:

`ArrayList` with the concentration amounts.

printCurrentConcentration

```
public void printCurrentConcentration(java.time.LocalDateTime time)
```

Description copied from interface: [IAction](#)

Prints concentration amount of doses at a specific time.

Specified by:

`printCurrentConcentration` in interface [IAction](#)

Parameters:

`time` - Specific time used to determine concentration amount.

getPeakConcentrationTime

```
public java.util.ArrayList<java.time.LocalDateTime> getPeakConcentrationTime(java.lang.Boolean includeTestDoses)
```

Description copied from interface: [IAction](#)

Time when the concentration is at its peak. It allows the user to choose if test doses will be displayed or not.

Specified by:

`getPeakConcentrationTime` in interface [IAction](#)

Parameters:

`includeTestDoses` - Determines whether test doses will be displayed.

Returns:

`ArrayList` containing peak concentration peak values of each dose.

printPeakConcentrationTime

```
public void printPeakConcentrationTime(java.lang.Boolean includeTestDoses)
```

Description copied from interface: [IAction](#)

Print concentration peaks. Includes dose information and its concentration peak.

Specified by:

`printPeakConcentrationTime` in interface [IAction](#)

Parameters:

`includeTestDoses` - Determines whether test doses will be displayed.

getWhenToDose

```
public java.util.ArrayList<java.time.LocalDateTime> getWhenToDose(java.lang.Double concentrationDesired)
```

Description copied from interface: [IAction](#)

Determines time when the patient should take next dose.

Specified by:

`getWhenToDose` in interface [IAction](#)

Parameters:

`concentrationDesired` - Dose amount desired sought.

Returns:

Time when the patient have to take next dose.

printWhenToDose

```
public void printWhenToDose(java.lang.Double amountDesired)
```

Description copied from interface: [IAction](#)

Prints when the patient should take next dose. Includes dose information and time to take next dose.

Specified by:

`printWhenToDose` in interface [IAction](#)

Parameters:

amountDesired - Dose amount desired sought.

saveFile

```
public void saveFile(java.lang.String filename)
```

Description copied from interface: [IAction](#)

Save Medicine to a file. By default, save directory is same directory as the application. Dose and Medicine MUST implement Serializable for this feature to work.

Specified by:

`saveFile` in interface [IAction](#)

Parameters:

filename - Name of save file to be saved.

loadFile

```
public void loadFile(java.lang.String filename)
```

Description copied from interface: [IAction](#)

Load a saved Medicine file. By default, load directory is same directory as the application. Dose and Medicine MUST implement Serializable for this feature to work.

Specified by:

`loadFile` in interface [IAction](#)

Parameters:

filename - Name of the file to be loaded

newFile

```
public void newFile(java.lang.String name,  
                   java.time.LocalDateTime tMax,  
                   java.time.LocalDateTime halfLife)
```

Description copied from interface: [IAction](#)

Create a new Medicine instance.

Specified by:

`newFile` in interface [IAction](#)

Parameters:

name - Name of medicine

tMax - TMax of medicine.

halfLife - Half life of medicine

printMedicine

```
public void printMedicine()
```

Description copied from interface: [IAction](#)

Print the name, tmax, and halfLife of the medicine.

Specified by:

`printMedicine` in interface [IAction](#)

removeTestDoses

```
public void removeTestDoses()
```

Description copied from interface: [IAction](#)

Removes all test doses.

Specified by:

`removeTestDoses` in interface [IAction](#)

getMedicine

```
public IMedicine getMedicine()
```

Description copied from interface: [IAction](#)

Retrieves the Medicine instance.

Specified by:

`getMedicine` in interface [IAction](#)

Returns:

Medicine instance

Class Controller

java.lang.Object
Controller

```
public class Controller
extends java.lang.Object
```

Field Summary

Fields	
Modifier and Type	Field and Description
(package private) static Action	action
(package private) static java.util.Scanner	userInput

Constructor Summary

Constructors	
Constructor and Description	
Controller()	

Method Summary

All Methods	Static Methods	Concrete Methods
Modifier and Type		Method and Description
private static void		checkIfInteger() Validates if user input is an integer only.
private static void		checkIfValidNumber() Validates if user input is a number, either Integer or Double.
private static void		clear() Clears the console screen.
private static IDose		createDose()

Creates Dose from user input.

```
private static java.time.LocalDateTime createLocalTime()
```

Generates LocalDateTime instance from user input.

```
private static void createMedicine()
```

Creates a Medicine instance from user input.

```
static void main(java.lang.String[] args)
```

```
private static void pause()
```

Pause scrolling for the console screen until user hits the Enter key.

```
private static void removeDose()
```

Removes dose using an index selected by the user.

```
private static void selectAction()
```

Display the list of actions that can be performed by the application.

```
private static void start()
```

Starts the program.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

userInput

```
static java.util.Scanner userInput
```

action

```
static Action action
```

Constructor Detail

Controller

```
public Controller()
```

Method Detail

createLocalTime

```
private static java.time.LocalDateTime createLocalTime()
```

Generates LocalTime instance from user input. Validates the user input to minimize application crash due to unexpected errors.

Returns:

LocalTime instance.

createMedicine

```
private static void createMedicine()
```

Creates a Medicine instance from user input.

createDose

```
private static IDose createDose()
```

Creates Dose from user input.

Returns:

A Dose instance

removeDose

```
private static void removeDose()
```

Removes dose using an index selected by the user.

checkIfInteger

```
private static void checkIfInteger()
```

Validates if user input is an integer only. If input is not an integer, the program stops.

checkIfValidNumber

```
private static void checkIfValidNumber()
```

Validates if user input is a number, either Integer or Double. If input is not a number, the program stops.

clear

```
private static void clear()
```

Clears the console screen.

pause

```
private static void pause()
```

Pause scrolling for the console screen until user hits the Enter key.

start

```
private static void start()
```

Starts the program. Display a welcome screen which allows the user to select whether to create a file or open an existing one.

selectAction

```
private static void selectAction()
```

Display the list of actions that can be performed by the application. The user must enter the corresponding number to execute the action.

main

```
public static void main(java.lang.String[] args)
```

[PACKAGE](#) **[CLASS](#)** [TREE](#) [DEPRECATED](#) [INDEX](#) [HELP](#)

[PREV CLASS](#) **[NEXT CLASS](#)** [FRAMES](#) [NO FRAMES](#) [ALL CLASSES](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#) [DETAIL: FIELD](#) | [CONSTR](#) | [METHOD](#)

Class Dose

java.lang.Object
Dose

All Implemented Interfaces:
IDose, java.io.Serializable

```
public class Dose
extends java.lang.Object
implements IDose, java.io.Serializable
```

See Also:
Serialized Form

Field Summary

Fields

Modifier and Type	Field and Description
private double	amount
private boolean	isTestDose
private java.time.LocalDateTime	timeTake

Constructor Summary

Constructors

Constructor and Description
Dose()
Dose(java.time.LocalDateTime timeTake, double amount)

Method Summary

All MethodsInstance MethodsConcrete Methods

Modifier and Type	Method and Description
-------------------	------------------------

void	createDose (java.time.LocalDateTime timeTake, double amount) Creates a Dose object.
double	getAmount () Concentration amount of the dose taken by patient.
java.time.LocalDateTime	getTimeTake () Time when the dose is taken by patient.
boolean	isTestDose () Identifies dose type.
void	setTestDose () Sets dose to test dose.
java.lang.String	toString ()

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Field Detail

timeTake

```
private java.time.LocalDateTime timeTake
```

amount

```
private double amount
```

isTestDose

```
private boolean isTestDose
```

Constructor Detail

Dose

```
public Dose()
```

Dose

```
public Dose(java.time.LocalTime timeTake,  
            double amount)
```

Method Detail

isTestDose

```
public boolean isTestDose()
```

Description copied from interface: [IDose](#)

Identifies dose type.

Specified by:

`isTestDose` in interface [IDose](#)

Returns:

True = TestDose (what-if dose); False = Dose (actual dose);

createDose

```
public void createDose(java.time.LocalTime timeTake,  
                       double amount)
```

Description copied from interface: [IDose](#)

Creates a Dose object.

Specified by:

`createDose` in interface [IDose](#)

Parameters:

`timeTake` - Time when the dose is taken by patient.

`amount` - Concentration amount of the dose taken by patient.

getAmount

```
public double getAmount()
```

Description copied from interface: [IDose](#)

Concentration amount of the dose taken by patient.

Specified by:

`getAmount` in interface [IDose](#)

Returns:

Concentration amount of Dose.

getTimeTake

```
public java.time.LocalDateTime getTimeTake()
```

Description copied from interface: [IDose](#)

Time when the dose is taken by patient.

Specified by:

`getTimeTake` in interface [IDose](#)

Returns:

`LocalTime` Time of Dose.

setTestDose

```
public void setTestDose()
```

Description copied from interface: [IDose](#)

Sets dose to test dose. Used for the what-if feature of the program.

Specified by:

`setTestDose` in interface [IDose](#)

toString

```
public java.lang.String toString()
```

Overrides:

`toString` in class `java.lang.Object`

[PACKAGE](#) **[CLASS](#)** [TREE](#) [DEPRECATED](#) [INDEX](#) [HELP](#)

[PREV CLASS](#) **[NEXT CLASS](#)** [FRAMES](#) [NO FRAMES](#) [ALL CLASSES](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#) [DETAIL: FIELD](#) | [CONSTR](#) | [METHOD](#)

Interface IAction

All Known Implementing Classes:

Action

```
public interface IAction
```

Method Summary

All Methods	Instance Methods	Abstract Methods
Modifier and Type		Method and Description
void		<code>addDose(IDose dose)</code> Adds a new dose.
java.util.ArrayList<java.lang.Double>		<code>getCurrentConcentration(java.time.LocalTime time)</code> Calculates concentration amount of doses at a specific time.
<code>I</code> Medicine		<code>getMedicine()</code> Retrieves the Medicine instance.
java.util.ArrayList<java.time.LocalTime>		<code>getPeakConcentrationTime(java.lang.Boolean includeTestDoses)</code> Time when the concentration is at its peak.
java.util.ArrayList<java.time.LocalTime>		<code>getWhenToDose(java.lang.Double concentrationDesired)</code> Determines time when the patient should take next dose.
void		<code>loadFile(java.lang.String filename)</code> Load a saved Medicine file.
void		<code>newFile(java.lang.String name, java.time.LocalTime tMax, java.time.LocalTime halfLife)</code> Create a new Medicine instance.
void		<code>printCurrentConcentration(java.time.LocalTime time)</code> Prints concentration amount of doses at a specific time.
void		<code>printDoses()</code> Prints all doses from Medicine's Dose array.
void		<code>printMedicine()</code> Print the name, tmax, and halfLife of the medicine.
void		<code>printPeakConcentrationTime(java.lang.Boolean includeTestDoses)</code> Print concentration peaks.
void		<code>printWhenToDose(java.lang.Double amountDesired)</code> Prints when the patient should take next dose.
void		<code>removeAllDoses()</code> Delete all doses from the dosages array.

void	<code>removeDose</code> (int index) Removes a dose from the dose from the doses array.
void	<code>removeTestDoses</code> () Removes all test doses.
void	<code>saveFile</code> (java.lang.String filename) Save Medicine to a file.

Method Detail

printDoses

```
void printDoses()
```

Prints all doses from Medicine's Dose array.

addDose

```
void addDose(IDose dose)
```

Adds a new dose. Added doses can be type Dose (actual dose) or TestDose (what-if dose)

Parameters:

dose - Dose object.

removeDose

```
void removeDose(int index)
```

Removes a dose from the dose from the doses array.

Parameters:

index - Index of the array to be erased

removeAllDoses

```
void removeAllDoses()
```

Delete all doses from the dosages array.

getPeakConcentrationTime

```
java.util.ArrayList<java.time.LocalDateTime> getPeakConcentrationTime(java.lang.Boolean includeTestDoses)
```

Time when the concentration is at its peak. It allows the user to choose if test doses will be displayed or not.

Parameters:

includeTestDoses - Determines whether test doses will be displayed.

Returns:

ArrayList containing peak concentration peak values of each dose.

printPeakConcentrationTime

```
void printPeakConcentrationTime(java.lang.Boolean includeTestDoses)
```

Print concentration peaks. Includes dose information and its concentration peak.

Parameters:

includeTestDoses - Determines whether test doses will be displayed.

getWhenToDose

```
java.util.ArrayList<java.time.LocalTime> getWhenToDose(java.lang.Double concentrationDesired)
```

Determines time when the patient should take next dose.

Parameters:

concentrationDesired - Dose amount desired sought.

Returns:

Time when the patient have to take next dose.

printWhenToDose

```
void printWhenToDose(java.lang.Double amountDesired)
```

Prints when the patient should take next dose. Includes dose information and time to take next dose.

Parameters:

amountDesired - Dose amount desired sought.

saveFile

```
void saveFile(java.lang.String filename)
```

Save Medicine to a file. By default, save directory is same directory as the application. Dose and Medicine MUST implement Serializable for this feature to work.

Parameters:

filename - Name of save file to be saved.

loadFile

```
void loadFile(java.lang.String filename)
```

Load a saved Medicine file. By default, load directory is same directory as the application. Dose and Medicine MUST implement Serializable for this feature to work.

Parameters:

filename - Name of the file to be loaded

newFile

```
void newFile(java.lang.String name,  
             java.time.LocalTime tMax,  
             java.time.LocalTime halfLife)
```

Create a new Medicine instance.

Parameters:

name - Name of medicine

tMax - TMax of medicine.

halfLife - Half life of medicine

printMedicine

```
void printMedicine()
```

Print the name, tmax, and halfLife of the medicine.

getCurrentConcentration

```
java.util.ArrayList<java.lang.Double> getCurrentConcentration(java.time.LocalDateTime time)
```

Calculates concentration amount of doses at a specific time.

Parameters:

time - Specific time used to determine concentration amount.

Returns:

ArrayList with the concentration amounts.

printCurrentConcentration

```
void printCurrentConcentration(java.time.LocalDateTime time)
```

Prints concentration amount of doses at a specific time.

Parameters:

time - Specific time used to determine concentration amount.

removeTestDoses

```
void removeTestDoses()
```

Removes all test doses.

getMedicine

```
IMedicine getMedicine()
```

Retrieves the Medicine instance.

Returns:

Medicine instance

Interface IDose

All Known Implementing Classes:

Dose

public interface IDose

Method Summary

All Methods	Instance Methods	Abstract Methods
Modifier and Type	Method and Description	
void	createDose (java.time.LocalDateTime timeTake, double amount) Creates a Dose object.	
double	getAmount () Concentration amount of the dose taken by patient.	
java.time.LocalDateTime	getTimeTake () Time when the dose is taken by patient.	
boolean	isTestDose () Identifies dose type.	
void	setTestDose () Sets dose to test dose.	

Method Detail

isTestDose
<pre>boolean isTestDose()</pre> <p>Identifies dose type.</p> <p>Returns:</p> <pre>True = TestDose (what-if dose); False = Dose (actual dose);</pre>
createDose

```
void createDose(java.time.LocalTime timeTake,  
               double amount)
```

Creates a Dose object.

Parameters:

timeTake - Time when the dose is taken by patient.

amount - Concentration amount of the dose taken by patient.

getTimeTake

```
java.time.LocalTime getTimeTake()
```

Time when the dose is taken by patient.

Returns:

LocalTime Time of Dose.

getAmount

```
double getAmount()
```

Concentration amount of the dose taken by patient.

Returns:

Concentration amount of Dose.

setTestDose

```
void setTestDose()
```

Sets dose to test dose. Used for the what-if feature of the program.

[PACKAGE](#) **[CLASS](#)** [TREE](#) [DEPRECATED](#) [INDEX](#) [HELP](#)

[PREV CLASS](#) [NEXT CLASS](#) [FRAMES](#) [NO FRAMES](#) [ALL CLASSES](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#) [DETAIL: FIELD](#) | [CONSTR](#) | [METHOD](#)

Interface IMedicine

All Known Implementing Classes:

Medicine

public interface **IMedicine**

Method Summary

All Methods	Instance Methods	Abstract Methods
Modifier and Type	Method and Description	
void	createMedicine (java.lang.String name, java.time.LocalDateTime tMax, java.time.LocalDateTime halfLife) Creates a medicine object.	
java.util.ArrayList<IDose>	getDoses () Corresponds to all existing doses.	
java.time.LocalDateTime	getHalfLife () Half life time of the medicine.	
java.lang.String	getName () Name of medicine.	
java.time.LocalDateTime	getTmax () Tmax of medicine	

Method Detail

createMedicine

```
void createMedicine(java.lang.String name,
                    java.time.LocalDateTime tMax,
                    java.time.LocalDateTime halfLife)
```

Creates a medicine object.

Parameters:

name - Name of medicine.

tMax - Time when medicine is at its peak concentration.
halfLife - Time required for medicine to decrease by half.

getDoses

```
java.util.ArrayList<IDose> getDoses()
```

Corresponds to all existing doses. Includes type Dose and TestDose.

Returns:

An array containing all doses.

getName

```
java.lang.String getName()
```

Name of medicine.

Returns:

Return name of the medicine.

getHalfLife

```
java.time.LocalDateTime getHalfLife()
```

Half life time of the medicine.

Returns:

Return the half life time of the medicine.

getTmax

```
java.time.LocalDateTime getTmax()
```

Tmax of medicine

Returns:

Return TMax of medicine.

Class Medicine

java.lang.Object
Medicine

All Implemented Interfaces:

IMedicine, java.io.Serializable

```
public class Medicine
  extends java.lang.Object
  implements IMedicine, java.io.Serializable
```

See Also:

Serialized Form

Field Summary

Fields	
Modifier and Type	Field and Description
private java.util.ArrayList<IDose>	doses
private java.time.LocalDateTime	halfLife
private java.lang.String	name
private java.time.LocalDateTime	tmax

Constructor Summary

Constructors	
Constructor and Description	
Medicine()	
Medicine (java.lang.String name, java.time.LocalDateTime tMax, java.time.LocalDateTime halfLife)	

Method Summary



All Methods**Instance Methods****Concrete Methods**

Modifier and Type	Method and Description
void	createMedicine (java.lang.String name, java.time.LocalDateTime tMax, java.time.LocalDateTime halfLife) Creates a medicine object.
java.util.ArrayList< IDose >	getDoses () Corresponds to all existing doses.
java.time.LocalDateTime	getHalfLife () Half life time of the medicine.
java.lang.String	getName () Name of medicine.
java.time.LocalDateTime	getTmax () Tmax of medicine
java.lang.String	toString ()

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Field Detail**name**

```
private java.lang.String name
```

tmax

```
private java.time.LocalDateTime tmax
```

halfLife

```
private java.time.LocalDateTime halfLife
```

doses

```
private java.util.ArrayList<IDose> doses
```

Constructor Detail

Medicine

```
public Medicine()
```

Medicine

```
public Medicine(java.lang.String name,  
                java.time.LocalDateTime tMax,  
                java.time.LocalDateTime halfLife)
```

Method Detail

createMedicine

```
public void createMedicine(java.lang.String name,  
                           java.time.LocalDateTime tMax,  
                           java.time.LocalDateTime halfLife)
```

Description copied from interface: [IMedicine](#)

Creates a medicine object.

Specified by:

`createMedicine` in interface [IMedicine](#)

Parameters:

`name` - Name of medicine.

`tMax` - Time when medicine is at its peak concentration.

`halfLife` - Time required for medicine to decrease by half.

getName

```
public java.lang.String getName()
```

Description copied from interface: [IMedicine](#)

Name of medicine.

Specified by:

`getName` in interface [IMedicine](#)

Returns:

Return name of the medicine.

getHalfLife

```
public java.time.LocalDateTime getHalfLife()
```

Description copied from interface: [IMedicine](#)

Half life time of the medicine.

Specified by:

`getHalfLife` in interface [IMedicine](#)

Returns:

Return the half life time of the medicine.

getDoses

```
public java.util.ArrayList<IDose> getDoses()
```

Description copied from interface: [IMedicine](#)

Corresponds to all existing doses. Includes type Dose and TestDose.

Specified by:

`getDoses` in interface [IMedicine](#)

Returns:

An array containing all doses.

getTmax

```
public java.time.LocalDateTime getTmax()
```

Description copied from interface: [IMedicine](#)

Tmax of medicine

Specified by:

`getTmax` in interface [IMedicine](#)

Returns:

Return TMax of medicine.

toString

```
public java.lang.String toString()
```

Overrides:

`toString` in class [java.lang.Object](#)

Package <Unnamed>

Interface Summary	
Interface	Description
IAction	
IDose	
IMedicine	

Class Summary	
Class	Description
Action	
Controller	
Dose	
Medicine	

Hierarchy For Package <Unnamed>

Class Hierarchy

- java.lang.Object
 - **Action** (implements **IAction**)
 - **Controller**
 - **Dose** (implements **IDose**, java.io.Serializable)
 - **Medicine** (implements **IMedicine**, java.io.Serializable)

Interface Hierarchy

- **IAction**
- **IDose**
- **IMedicine**

Deprecated API

Contents

A C D G H I L M N P R S T U

A

Action - Class in <Unnamed>

Action() - Constructor for class Action

action - Static variable in class Controller

addDose(IDose) - Method in class Action

addDose(IDose) - Method in interface IAction

 Adds a new dose.

amount - Variable in class Dose

A C D G H I L M N P R S T U

How This API Document Is Organized

This API (Application Programming Interface) document has pages corresponding to the items in the navigation bar, described as follows.

Package

Each package has a page that contains a list of its classes and interfaces, with a summary for each. This page can contain six categories:

- Interfaces (*italic*)
- Classes
- Enums
- Exceptions
- Errors
- Annotation Types

Class/Interface

Each class, interface, nested class and nested interface has its own separate page. Each of these pages has three sections consisting of a class/interface description, summary tables, and detailed member descriptions:

- Class inheritance diagram
- Direct Subclasses
- All Known Subinterfaces
- All Known Implementing Classes
- Class/interface declaration
- Class/interface description
- Nested Class Summary
- Field Summary
- Constructor Summary
- Method Summary
- Field Detail
- Constructor Detail
- Method Detail

Each summary entry contains the first sentence from the detailed description for that item. The summary entries are alphabetical, while the detailed descriptions are in the order they appear in the source code. This preserves the logical groupings established by the programmer.

Annotation Type

Each annotation type has its own separate page with the following sections:

- Annotation Type declaration
- Annotation Type description
- Required Element Summary
- Optional Element Summary

- [Element Detail](#)

Enum

Each enum has its own separate page with the following sections:

- Enum declaration
- Enum description
- Enum Constant Summary
- Enum Constant Detail

Tree (Class Hierarchy)

There is a [Class Hierarchy](#) page for all packages, plus a hierarchy for each package. Each hierarchy page contains a list of classes and a list of interfaces. The classes are organized by inheritance structure starting with `java.lang.Object`. The interfaces do not inherit from `java.lang.Object`.

- When viewing the Overview page, clicking on "Tree" displays the hierarchy for all packages.
- When viewing a particular package, class or interface page, clicking "Tree" displays the hierarchy for only that package.

Deprecated API

The [Deprecated API](#) page lists all of the API that have been deprecated. A deprecated API is not recommended for use, generally due to improvements, and a replacement API is usually given. Deprecated APIs may be removed in future implementations.

Index

The [Index](#) contains an alphabetic list of all classes, interfaces, constructors, methods, and fields.

Prev/Next

These links take you to the next or previous class, interface, package, or related page.

Frames/No Frames

These links show and hide the HTML frames. All pages are available with or without frames.

All Classes

The [All Classes](#) link shows all classes and interfaces except non-static nested types.

Serialized Form

Each serializable or externalizable class has a description of its serialization fields and methods. This information is of interest to re-implementors, not to developers using the API. While there is no link in the navigation bar, you can get to this information by going to any serialized class and clicking "Serialized Form" in the "See also" section of the class description.

Constant Field Values

The [Constant Field Values](#) page lists the static final fields and their values.

This help file applies to API documentation generated using the standard doclet.

All Classes

- Action
- Controller
- Dose
- IAction*
- IDose*
- IMedicine*
- Medicine

Serialized Form

Package <Unnamed>

Class *Dose* extends *java.lang.Object* implements *Serializable*

Serialized Fields

timeTake

```
java.time.LocalDateTime timeTake
```

amount

```
double amount
```

isTestDose

```
boolean isTestDose
```

Class *Medicine* extends *java.lang.Object* implements *Serializable*

Serialized Fields

name

```
java.lang.String name
```

tmax

```
java.time.LocalDateTime tmax
```

halfLife

```
java.time.LocalDateTime halfLife
```

doses

```
java.util.ArrayList<E> doses
```

[PACKAGE](#) [CLASS](#) [TREE](#) [DEPRECATED](#) [INDEX](#) [HELP](#)

[PREV](#) [NEXT](#) [FRAMES](#) [NO FRAMES](#) [ALL CLASSES](#)

Hierarchy For All Packages

Class Hierarchy

- java.lang.Object
 - **Action** (implements **IAction**)
 - **Controller**
 - **Dose** (implements **IDose**, java.io.Serializable)
 - **Medicine** (implements **IMedicine**, java.io.Serializable)

Interface Hierarchy

- **IAction**
- **IDose**
- **IMedicine**

A C D G H I L M N P R S T U

C

- checkIfInteger()** - Static method in class [Controller](#)
Validates if user input is an integer only.
- checkIfValidNumber()** - Static method in class [Controller](#)
Validates if user input is a number, either [Integer](#) or [Double](#).
- clear()** - Static method in class [Controller](#)
Clears the console screen.
- Controller** - Class in [<Unnamed>](#)
- Controller()** - Constructor for class [Controller](#)
- createDose()** - Static method in class [Controller](#)
Creates [Dose](#) from user input.
- createDose(LocalTime, double)** - Method in class [Dose](#)
- createDose(LocalTime, double)** - Method in interface [IDose](#)
Creates a [Dose](#) object.
- createLocalTime()** - Static method in class [Controller](#)
Generates [LocalTime](#) instance from user input.
- createMedicine()** - Static method in class [Controller](#)
Creates a [Medicine](#) instance from user input.
- createMedicine(String, LocalTime, LocalTime)** - Method in interface [IMedicine](#)
Creates a medicine object.
- createMedicine(String, LocalTime, LocalTime)** - Method in class [Medicine](#)

A C D G H I L M N P R S T U

A C D G H I L M N P R S T U

D

Dose - Class in <Unnamed>

Dose() - Constructor for class Dose

Dose(LocalTime, double) - Constructor for class Dose

doses - Variable in class Medicine

A C D G H I L M N P R S T U

A C D G H I L M N P R S T U

G

getAmount() - Method in class [Dose](#)

getAmount() - Method in interface [IDose](#)

Concentration amount of the dose taken by patient.

getCurrentConcentration(LocalTime) - Method in class [Action](#)

getCurrentConcentration(LocalTime) - Method in interface [IAction](#)

Calculates concentration amount of doses at a specific time.

getDoses() - Method in interface [IMedicine](#)

Corresponds to all existing doses.

getDoses() - Method in class [Medicine](#)

getHalfLife() - Method in interface [IMedicine](#)

Half life time of the medicine.

getHalfLife() - Method in class [Medicine](#)

getMedicine() - Method in class [Action](#)

getMedicine() - Method in interface [IAction](#)

Retrieves the Medicine instance.

getName() - Method in interface [IMedicine](#)

Name of medicine.

getName() - Method in class [Medicine](#)

getPeakConcentrationTime(Boolean) - Method in class [Action](#)

getPeakConcentrationTime(Boolean) - Method in interface [IAction](#)

Time when the concentration is at its peak.

getTimeTake() - Method in class [Dose](#)

getTimeTake() - Method in interface [IDose](#)

Time when the dose is taken by patient.

getTmax() - Method in interface [IMedicine](#)

Tmax of medicine

getTmax() - Method in class [Medicine](#)

getWhenToDose(Double) - Method in class [Action](#)

getWhenToDose(Double) - Method in interface [IAction](#)

Determines time when the patient should take next dose.

A C D G H I L M N P R S T U

A C D G H I L M N P R S T U

H

halfLife - Variable in class Medicine

A C D G H I L M N P R S T U

A C D G H I L M N P R S T U

I

IAction - Interface in <Unnamed>

IDose - Interface in <Unnamed>

IMedicine - Interface in <Unnamed>

isTestDose - Variable in class Dose

isTestDose() - Method in class Dose

isTestDose() - Method in interface IDose
Identifies dose type.

A C D G H I L M N P R S T U

A C D G H I L M N P R S T U

L

loadFile(String) - Method in class *Action*

loadFile(String) - Method in interface *IAction*

Load a saved Medicine file.

A C D G H I L M N P R S T U

A C D G H I L M N P R S T U

M

- main(String[])** - Static method in class [Controller](#)
- medicine** - Variable in class [Action](#)
- Medicine** - Class in [<Unnamed>](#)
- Medicine()** - Constructor for class [Medicine](#)
- Medicine(String, LocalTime, LocalTime)** - Constructor for class [Medicine](#)

A C D G H I L M N P R S T U

A C D G H I L M N P R S T U

N

- name** - Variable in class Medicine
- newFile(String, LocalTime, LocalTime)** - Method in class Action
- newFile(String, LocalTime, LocalTime)** - Method in interface IAction
 - Create a new Medicine instance.

A C D G H I L M N P R S T U

A C D G H I L M N P R S T U

P

- pause()

- Static method in class Controller

Pause scrolling fot the console screen until user hits the Enter key.
- printCurrentConcentration(LocalTime)

- Method in class Action
- printCurrentConcentration(LocalTime)

- Method in interface IAction

Prints concentration amount of doses at a specific time.
- printDoses()

- Method in class Action
- printDoses()

- Method in interface IAction

Prints all doses from Medicine's Dose array.
- printMedicine()

- Method in class Action
- printMedicine()

- Method in interface IAction

Print the name, tmax, and halfLife of the medicine.
- printPeakConcentrationTime(Boolean)

- Method in class Action
- printPeakConcentrationTime(Boolean)

- Method in interface IAction

Print concentration peaks.
- printWhenToDose(Double)

- Method in class Action
- printWhenToDose(Double)

- Method in interface IAction

Prints when the patient should take next dose.

A C D G H I L M N P R S T U

A C D G H I L M N P R S T U

R

removeAllDoses() - Method in class [Action](#)

removeAllDoses() - Method in interface [IAction](#)

Delete all doses from the dosages array.

removeDose(int) - Method in class [Action](#)

removeDose() - Static method in class [Controller](#)

Removes dose using an index selected by the user.

removeDose(int) - Method in interface [IAction](#)

Removes a dose from the dose from the doses array.

removeTestDoses() - Method in class [Action](#)

removeTestDoses() - Method in interface [IAction](#)

Removes all test doses.

A C D G H I L M N P R S T U

A C D G H I L M N P R S T U

S

saveFile(String) - Method in class [Action](#)

saveFile(String) - Method in interface [IAction](#)

Save Medicine to a file.

selectAction() - Static method in class [Controller](#)

Display the list of actions that can be performed by the application.

setTestDose() - Method in class [Dose](#)

setTestDose() - Method in interface [IDose](#)

Sets dose to test dose.

start() - Static method in class [Controller](#)

Starts the program.

A C D G H I L M N P R S T U

A C D G H I L M N P R S T U

T

- timeTake** - Variable in class Dose
- tmax** - Variable in class Medicine
- toString()** - Method in class Dose
- toString()** - Method in class Medicine

A C D G H I L M N P R S T U

A C D G H I L M N P R S T U

U

userInput - Static variable in class Controller

A C D G H I L M N P R S T U

Constant Field Values

Contents