## **Data dictionary**

Name	Туре	Protection	Rationale	Description
Vector			Vector was a class designed to work as an Array that would rezise as needed	Array that stores unlimited number of a generic type
List*	elemType	-	List of elements of that array, it's a pointer as this is the way Vectors work	This is a list of a certain generic type
Length	Int	-	Size of the Array Vector necessary for the structure of the array	This is the current size of the array Vector
maxSize	Int	-	The current max size of the array that is changed when the array reach the capacity	This is the max size of the array Vector
Vector(int)	Vector	+	Vector constructor that can be used to initialize a Vector with an specific type	Standard constructor of Vector takes a interger as paramenter
Vector	Vector	~	Vector destructor	Vector destructor
insertEnd(const elemType)	elemType	+	This function insert elements in the end of the stack	Insert a generic element to the end of the Vector
retrieveAt(int, elemType)	elemType	+	This is necessary to pick a certain element of the vector array	Gets a index position and return the element of the array Vector
Resize(int)	void	+	This function doubles the size of the array when the array matches the length	Function that doubles the size of vector when it reaches it capacity
Metadata			This class was created to be an object that holds all registers of a specific weather data collection	Class that works as an object to hold the variables of the csv lines
m_windSpeed	Double	-	This is a private variable that is used to hold the speed in Km/h	Holds the wind speed max for a specific register
M_solarRad	Double	-	This variable holds the solar radiation in W/m2	Holds the solar radiation registered for a sigle register

M_dateObject	Date	-	Object of type date that	Variable that is an
			will be to hold a date	object of type date

			while a read the file csv	
Metadata()	Metadata	+	Constructor for metadata, don't have paramenters, used in case just need to create an object without passing the values	Default constructor for metadata class
Metadata(Date, Time, double, double)	Metadata	+	Constructor that holds parameters	Constructor of Metadata that take Parameters
getWindSpeed()	Double	+	Function that returns wind speed for a specific register	Function that returns wind speed for a specific register
getSolarRad()	Double	+	Function that returns solar radiation from a register	Function that returns solar radiation from a register
getDate()	Date	+	Returns a date of a weather register	Returns a date of a weather register
getTime()	Time	+	Returns time of a specific weather event	Returns time of a specific weather event
setWindSpeed(do uble)	Void	+	Set the wind speed of a register Metadata	Set the wind speed of a register Metadata
setSolarRad(dou ble)	Void	+	Set solar radiation of a metadata register	Set solar radiation of a metadata register
setDate(Date)	Void	+	Set the date of a metadata register	Set the date of a metadata register
setTime(Time)	Void	+	Set the time of a metadata Register	Set the time of a metadata Register
Time			Class that collects time of a metadata register	Class that collects time of a metadata register
M_hour	Int	-	Variable that holds the hour of a register	Variable that holds the hour of a register
M_minute	Int	-	Variable that holds the minutes of a register	Variable that holds the minutes of a register
Time()	Time	+	Default constructor of class Time	Default constructor of class Time
Time(int, int)	Time	+	Constructor of Time object that takes parameters	Constructor of Time object that takes parameters
getHour()	Int	+	Returns an hour of an event	Returns an hour of an event
getMinute()	Int	+	Returns a minute of an event	Returns a minute of an event
setHour(int)	Void	+	Set an hour of a register	Set an hour of a register

setMinute(int)	Void	+	Set the minute of a	Set the minute of a
			regiter	regiter
Date			Class date that gets a set	Class date that
			of day, month and year	gets a set of day,
				month and year
M_day	Int	-	Day variable is the day of	Day variable is the
			a register	day of a register
M_month	Int	-	Month of a register	Month of a register
M_year	Int	-	Year of a register	Year of a register
			metadata	metadata
Date()	Date	+	Default constructor of	Default constructor
,			Date object	of Date object
Date(int, int, int)	Date	+	Date constructor that	Date constructor
			takes parameters	that takes
				parameters
getDay()	Int	+	Function that returns the	Function that
			day of a register	returns the day of
				a register
getMonth()	Int	+	Function that returns the	Function that
			month of an event	returns the month
				of an event
getYear()	Int	+	Return the year	Return the year
setDay(int)	Void	+	Set the day of a specific	Set the day of a
			register	specific register
setMonth(int)	Void	+	Set the month of a	Set the month of a
			metadata register	metadata register
setYear(int)	Void	+	Set the year of a register	Set the year of a
				register

Weather			This class works on behalf of the client class; it should have a private member variable of Vector <metadata>, which would give access to the data (stored in the vector, required for processing).</metadata>	the processing required is done in this class – not in the client class
m_count	int	-	used to count all register read in all files	used to count all register read in all files
metadataList	Vector <metadata></metadata>	-	used to store all objects metadata in the calculations in funtions 1 to 4array	used to store all objects metadata in the calculations in funtions 1 to 4
metadataTree	Int	-	The current max size of the array that is changed when the array reach the	This is the max size of the array Vector

			capacity	
timeTree	Vector	+	Vector constructor that can be used to initialize a Vector with an specific type	Standard constructor of Vector takes a interger as paramenter
yearCounter	Vector	~	Vector destructor	Vector destructor
monthCounter	elemType	+	This function insert elements in the end of the stack	Insert a generic element to the end of the Vector
windMaxDouble	elemType	+	This is necessary to pick a certain element of the vector array	Gets a index position and return the element of the array Vector
solarRadDouble	void	+	This function doubles the size of the array when the array matches the length	Function that doubles the size of vector when it reaches it capacity
binaryTreeType			The Binary Tree class is a Binary Search Tree that works as a data structure which would not store repeated values and have faster access to its ccomponents thanks to the way the algorithms are stored	The Binary Tree class is a Binary Search Tree that works as a data structure which would not store repeated values and have faster access to its ccomponents thanks to the way the algorithms are stored
nodeType	Double	-	This is a private variable that is used to hold the speed in Km/h	Holds the wind speed max for a specific register
info	Double	-	This variable holds the solar radiation in W/m2	Holds the solar radiation registered for a sigle register

ILink	Date	-	Object of type date that will be to hold a date while a read the file csv	Variable that is an object of type date
rLink	Metadata	+	Constructor for metadata, don't have paramenters, used in case just need to create an object without passing the values	Default constructor for metadata class
root	Metadata	+	Constructor that holds parameters	Constructor of Metadata that take

						Parameters
isEmpty()	Double	+			ction that returns	Function that
				wind	d speed for a specific	returns wind speed
				regi	ster	for a specific
						register
inorderTraversal()	Double	+			ction that returns	Function that
					r radiation from a	returns solar
				regi	ster	radiation from a
					<u> </u>	register
preorderTravers					This class works on	the processing
al()					behalf of the client class;	
					it should have a private	this class – not in
					member variable of	the client class
					Vector <metadata>,</metadata>	
					which would give access to the data (stored in the	
					vector, required for	
					processing).	
postorderTravers	int		_		used to count all	used to count all
alDelete()					register read in all files	
alboioto()					rogicior rodd iir dii iiioo	all files
postorderTravers	Vector <metac< td=""><td>data&gt;</td><td>-</td><td></td><td>used to store all</td><td>used to store all</td></metac<>	data>	-		used to store all	used to store all
al()					objects metadata in	objects metadata
V					the calculations in	in the
					funtions 1 to 4array	calculations in
					•	funtions 1 to 4
retrieve(const	Int		-		The current max size	This is the max
elemType&					of the array that is	size of the array
searchItem)					changed when the	Vector
					array reach the	
					capacity	
search(const	Vector		+		Vector constructor	Standard
elemType&					that can be used to	constructor of
searchItem)					initialize a Vector with	Vector takes a
					an specific type	interger as
insertToTree(nod	Vector		~		Vector destructor	paramenter Vector destructor
eType*	Vector		~		vector destructor	vector destructor
newNode,						
nodeType*						
parent);						
insert(elemType	elemType		+		This function insert	Insert a generic
insertItem);	7,5				elements in the end of	element to the
,					the stack	end of the Vector
deleteNode(cons	elemType		+		This is necessary to	Gets a index
t elemType&					pick a certain element	position and
deleteltem);					of the vector array	return the
						element of the
						array Vector
binaryTreeType()	void		+		This function doubles	Function that
1:					the size of the array	doubles the size
,					when the array	of vector when it

			matches the length	reaches it capacity
binaryTreeType			The Binary Tree class is a Binary Search Tree that works as a data structure which would not store repeated values and have faster access to its ccomponents thanks to the way the algorithms are stored	
root	nodeType	-	elem type that will store value that will be stored in the tree	elem type that will store value that will be stored in the tree C:\Users\327839 92\Desktop\ASSI GNMENT1\ASSI GNMENT1.cbp
info	Double	-	This variable holds the solar radiation in W/m2	
ILink	Date	-	Object of type date that	Variable that is an
			will be to hold a date	object of type date

ILink	Date	-	Object of type date that	Variable that is an
			will be to hold a date	object of type date
			while a read the file csv	
rLink	Metadata	+	Constructor for metadata,	Default constructor
			don't have paramenters,	for metadata class
			used in case just need to	
			create an object without	
	1		passing the values	
root	Metadata	+	Constructor that holds	Constructor of
			parameters	Metadata that take
				Parameters
isEmpty()	Double	+	Function that returns	Function that
			wind speed for a specific	returns wind speed
			register	for a specific
				register
inorderTraversal()	Double	+	Function that returns	Function that
			solar radiation from a	returns solar
			register	radiation from a
				register
preorderTraversal	Date	+	Returns a date of a	Returns a date of a
()			weather register	weather register
postorderTravers	Time	+	Returns time of a specific	Returns time of a
alDelete()			weather event	specific weather
				event

postorderTravers	Void	+	Set the wind speed of a	Set the wind speed
al()	Void	"	register Metadata	of a register
αι()			register ivioladata	Metadata
retrieve(const	Void	+	Set solar radiation of a	Set solar radiation
elemType&	Void	'	metadata register	of a metadata
searchItem)			metadata register	register
const;				rogiotor
search(const	Void	+	Set the date of a	Set the date of a
elemType&	Vola		metadata register	metadata register
searchItem)			motadata rogioto.	motadata rogiotor
const;				
insertToTree(nod	Void	+	Set the time of a	Set the time of a
eType* newNode,			metadata Register	metadata Register
nodeType*			motorasta regional	motadata riogisto.
parent);				
insert(elemType			Class that collects time of	Class that collects
insertItem);			a metadata register	time of a metadata
,,			a the same to grow	register
deleteNode(const	Int	-	Variable that holds the	Variable that holds
elemType&			hour of a register	the hour of a
deleteItem);				register
binaryTreeType();	Int	-	Variable that holds the	Variable that holds
, , , , , , , , , , , , , , , , , , , ,			minutes of a register	the minutes of a
			3	register
~binaryTreeType(	Time	+	Default constructor of	Default constructor
);			class Time	of class Time
inorder(nodeType	Time	+	Constructor of Time	Constructor of
*p) const;			object that takes	Time object that
,			parameters	takes parameters
preorder(nodeTy	Int	+	Returns an hour of an	Returns an hour of
pe *p) const;			event	an event
getMinute()	Int	+	Returns a minute of an	Returns a minute
			event	of an event
postorder(nodeTy	Void	+	Set an hour of a register	Set an hour of a
pe *p) const;				register
deleteFromTree(	Void	+	Set the minute of a	Set the minute of a
nodeType* &p);			regiter	regiter
Date			Class date that gets a set	Class date that
			of day, month and year	gets a set of day,
				month and year
M_day	Int	-	Day variable is the day of	Day variable is the
			a register	day of a register
M_month	Int	-	Month of a register	Month of a register
M_year	Int	-	Year of a register	Year of a register
			metadata	metadata
Date()	Date	+	Default constructor of	Default constructor
			Date object	of Date object
Date(int, int, int)	Date	+	Date constructor that	Date constructor
			takes parameters	that takes
				parameters
getDay()	Int	+	Function that returns the	Function that
			day of a register	returns the day of

				a register
getMonth()	Int	+	Function that returns the	Function that
			month of an event	returns the month
				of an event
getYear()	Int	+	Return the year	Return the year
setDay(int)	Void	+	Set the day of a specific	Set the day of a
			register	specific register
setMonth(int)	Void	+	Set the month of a	Set the month of a
			metadata register	metadata register
setYear(int)	Void	+	Set the year of a register	Set the year of a
, ,				register