

Lukos

Abricot 0.3

Généré par Doxygen 1.8.18

1 GNU GENERAL PUBLIC LICENSE	1
1.0.1 Preamble	1
1.0.2 TERMS AND CONDITIONS	2
1.0.2.1 0. Definitions.	2
1.0.2.2 1. Source Code.	2
1.0.2.3 2. Basic Permissions.	3
1.0.2.4 3. Protecting Users' Legal Rights From Anti-Circumvention Law.	3
1.0.2.5 4. Conveying Verbatim Copies.	3
1.0.2.6 5. Conveying Modified Source Versions.	3
1.0.2.7 6. Conveying Non-Source Forms.	4
1.0.2.8 7. Additional Terms.	5
1.0.2.9 8. Termination.	5
1.0.2.10 9. Acceptance Not Required for Having Copies.	6
1.0.2.11 10. Automatic Licensing of Downstream Recipients.	6
1.0.2.12 11. Patents.	6
1.0.2.13 12. No Surrender of Others' Freedom.	7
1.0.2.14 13. Use with the GNU Affero General Public License.	7
1.0.2.15 14. Revised Versions of this License.	7
1.0.2.16 15. Disclaimer of Warranty.	8
1.0.2.17 16. Limitation of Liability.	8
1.0.2.18 17. Interpretation of Sections 15 and 16.	8
2 Index des espaces de nommage	9
2.1 Liste des espaces de nommage	9
3 Index hiérarchique	11
3.1 Hiérarchie des classes	11
4 Index des classes	13
4.1 Liste des classes	13
5 Index des fichiers	15
5.1 Liste des fichiers	15
6 Documentation des espaces de nommage	17
6.1 Référence de l'espace de nommage Ui	17
6.1.1 Description détaillée	17
7 Documentation des classes	19
7.1 Référence de la classe Budget	19
7.1.1 Description détaillée	20
7.1.2 Documentation des fonctions membres	20
7.1.2.1 addOperation()	20
7.1.2.2 getBalance()	21

7.1.2.3	getDiffBtwnBalances()	21
7.1.2.4	getForecastBalance()	21
7.1.2.5	load()	21
7.1.2.6	operator<() [1/2]	22
7.1.2.7	operator<() [2/2]	22
7.1.2.8	save()	22
7.1.2.9	setEdit()	23
7.1.2.10	show()	23
7.2	Référence de la classe BudgetCategory	23
7.2.1	Description détaillée	25
7.2.2	Documentation des fonctions membres	25
7.2.2.1	addOperation()	25
7.2.2.2	delOperation()	25
7.2.2.3	getColor()	26
7.2.2.4	getSumBalances()	26
7.2.2.5	getType()	26
7.2.2.6	load()	26
7.2.2.7	save() [1/2]	27
7.2.2.8	save() [2/2]	27
7.2.2.9	setColor()	27
7.2.2.10	setOccurence()	28
7.2.2.11	setSubject()	28
7.2.2.12	setType()	28
7.2.2.13	show()	28
7.3	Référence de la classe Calendar	29
7.3.1	Description détaillée	30
7.3.2	Documentation des fonctions membres	30
7.3.2.1	addBudgetCategory()	30
7.3.2.2	getBalance()	31
7.3.2.3	getInstance()	31
7.3.2.4	load()	31
7.3.2.5	operator>>()	31
7.3.2.6	remove()	32
7.3.2.7	save()	32
7.3.2.8	show()	32
7.3.3	Documentation des données membres	33
7.3.3.1	calendar	33
7.4	Référence du modèle de la classe Container< T >	33
7.4.1	Description détaillée	34
7.4.2	Documentation des constructeurs et destructeur	34
7.4.2.1	Container()	34
7.4.3	Documentation des fonctions membres	34

7.4.3.1 begin()	35
7.4.3.2 getSize()	35
7.4.3.3 insert()	35
7.4.3.4 operator[]() [1/2]	35
7.4.3.5 operator[]() [2/2]	36
7.4.3.6 remove() [1/2]	36
7.4.3.7 remove() [2/2]	36
7.5 Référence de la classe Date	37
7.5.1 Description détaillée	38
7.5.2 Documentation des fonctions membres	38
7.5.2.1 fromString()	38
7.5.2.2 getDay()	38
7.5.2.3 getMonth()	39
7.5.2.4 getToday()	39
7.5.2.5 getYear()	39
7.5.2.6 operator+()	39
7.5.2.7 operator-()	40
7.5.2.8 operator<() [1/2]	40
7.5.2.9 operator<() [2/2]	40
7.5.2.10 toQDate()	41
7.5.2.11 toString()	41
7.6 Référence de la classe Event	41
7.6.1 Description détaillée	42
7.6.2 Documentation des fonctions membres	42
7.6.2.1 getDate()	42
7.6.2.2 getFrequency()	42
7.6.2.3 getOccurence()	43
7.6.2.4 getSubject()	43
7.6.2.5 operator<() [1/2]	43
7.6.2.6 operator<() [2/2]	43
7.6.2.7 operator>() [1/2]	44
7.6.2.8 operator>() [2/2]	44
7.7 Référence de la classe Exception	45
7.7.1 Description détaillée	45
7.7.2 Documentation des fonctions membres	45
7.7.2.1 getStdString()	45
7.7.2.2 getString()	45
7.8 Référence de la classe Frequency	46
7.8.1 Description détaillée	46
7.8.2 Documentation des constructeurs et destructeur	46
7.8.2.1 Frequency()	47
7.8.3 Documentation des fonctions membres	47

7.8.3.1 fromString()	47
7.8.3.2 getFrequency()	47
7.8.3.3 getNbMonth()	48
7.8.3.4 getNbWeek()	48
7.8.3.5 getNbYear()	48
7.8.3.6 getRythm()	48
7.8.3.7 operator*()	48
7.8.3.8 toString()	49
7.9 Référence de la classe Container< T >::Iterator	49
7.9.1 Description détaillée	49
7.9.2 Documentation des fonctions membres	50
7.9.2.1 isTerminated()	50
7.9.2.2 operator*()	50
7.9.2.3 operator++()	50
7.10 Référence de la classe Operation	50
7.10.1 Description détaillée	51
7.10.2 Documentation des fonctions membres	51
7.10.2.1 getDate()	52
7.10.2.2 getDescription()	52
7.10.2.3 getReference()	52
7.10.2.4 getSubject()	52
7.10.2.5 getValue()	52
7.10.2.6 load()	52
7.10.2.7 operator<()	53
7.10.2.8 save()	53
7.11 Référence de la classe Parameters	53
7.11.1 Description détaillée	55
7.11.2 Documentation des constructeurs et destructeur	55
7.11.2.1 Parameters()	55
7.11.3 Documentation des fonctions membres	55
7.11.3.1 changeDevise	55
7.11.3.2 changeDeviseVisible	55
7.11.3.3 changeFormatDate	56
7.11.3.4 changeMaxPoints	56
7.11.3.5 changeWritePDF	56
7.11.3.6 changeWriteRegistry	57
7.11.3.7 changeWriteSave	57
7.11.3.8 getAppLongName()	57
7.11.3.9 getAppName()	57
7.11.3.10 getAppVersion()	58
7.11.3.11 getDevise()	58
7.11.3.12 getDeviseVisible()	58

7.11.3.13 getDirPDF()	58
7.11.3.14 getDirRegistry()	58
7.11.3.15 getFileSave()	59
7.11.3.16 getFormatDate()	59
7.11.3.17 getGraphicsMaxPoints()	59
7.11.3.18 openHelp()	59
7.12 Référence de la classe Ui::Principale	59
7.13 Référence de la classe Principale	60
7.13.1 Documentation des constructeurs et destructeur	61
7.13.1.1 Principale()	61
7.14 Référence de la classe QCustomFrame	61
7.14.1 Description détaillée	62
7.14.2 Documentation des constructeurs et destructeur	62
7.14.2.1 QCustomFrame()	63
7.14.3 Documentation des fonctions membres	63
7.14.3.1 focusOn	63
7.14.3.2 setBudgetCategory()	63
7.14.3.3 setMaxPoints()	64
7.15 Référence de la structure qt_meta_stringdata_Parameters_t	64
7.16 Référence de la structure qt_meta_stringdata_Principale_t	64
7.17 Référence de la structure qt_meta_stringdata_QCustomFrame_t	64
7.18 Référence de la classe Ui_Principale	65
8 Documentation des fichiers	69
8.1 Référence du fichier C:/Users/Jérémy/Desktop/projets/Budgettiseur/Budgettiseur/header/Calendar.h	69
8.1.1 Description détaillée	69
8.1.2 Documentation des fonctions	70
8.1.2.1 getAttributeFromXML()	70
8.1.2.2 operator<<() [1/3]	70
8.1.2.3 operator<<() [2/3]	71
8.1.2.4 operator<<() [3/3]	71
8.2 Référence du fichier C:/Users/Jérémy/Desktop/projets/Budgettiseur/Budgettiseur/header/Container.h	72
8.2.1 Description détaillée	72
8.3 Référence du fichier C:/Users/Jérémy/Desktop/projets/Budgettiseur/Budgettiseur/header/Event.h	72
8.3.1 Description détaillée	73
8.3.2 Documentation des fonctions	73
8.3.2.1 operator<<() [1/2]	73
8.3.2.2 operator<<() [2/2]	74
8.4 Référence du fichier C:/Users/Jérémy/Desktop/projets/Budgettiseur/Budgettiseur/header/Exception.h	74
8.4.1 Description détaillée	75
8.5 Référence du fichier C:/Users/Jérémy/Desktop/projets/Budgettiseur/Budgettiseur/header/Parameters.h	75
8.5.1 Description détaillée	76

Chapitre 1

GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <https://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

1.0.1 Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program—to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

1.0.2 TERMS AND CONDITIONS

1.0.2.1 0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1.0.2.2 1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

1.0.2.3 2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

1.0.2.4 3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

1.0.2.5 4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

1.0.2.6 5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
- d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

1.0.2.7 6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

- a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
- b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.
- c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.
- d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.
- e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

1.0.2.8 7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

1.0.2.9 8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

1.0.2.10 9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

1.0.2.11 10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

1.0.2.12 11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

1.0.2.13 12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

1.0.2.14 13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

1.0.2.15 14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

1.0.2.16 15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

1.0.2.17 16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

1.0.2.18 17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

Chapitre 2

Index des espaces de nommage

2.1 Liste des espaces de nommage

Liste de tous les espaces de nommage documentés avec une brève description:

Ui	17
----	----

Chapitre 3

Index hiérarchique

3.1 Hiérarchie des classes

Cette liste d'héritage est classée approximativement par ordre alphabétique :

Container< T >	33
Container< Budget * >	33
BudgetCategory	23
Container< BudgetCategory * >	33
Calendar	29
Container< Operation * >	33
Budget	19
Date	37
Event	41
BudgetCategory	23
Exception	45
Frequency	46
Container< T >::Iterator	49
Operation	50
QFrame	
QCustomFrame	61
QMainWindow	
Principale	60
QObject	
Parameters	53
qt_meta_stringdata_Parameters_t	64
qt_meta_stringdata_Principale_t	64
qt_meta_stringdata_QCustomFrame_t	64
Ui_Principale	65
Ui::Principale	59

Chapitre 4

Index des classes

4.1 Liste des classes

Liste des classes, structures, unions et interfaces avec une brève description :

Budget	
Class Budget Représente un budget	19
BudgetCategory	
Class BudgetCategory Représente une catégorie budgétaire contenant plusieurs budgets . . .	23
Calendar	
Class Calendar Représente un calendrier budgétaire contenant plusieurs catégories budgétaires	29
Container< T >	
Class Container<T> Représente un contenant d'objets T	33
Date	
Class Date Représente une date	37
Event	
Class Event Représente un événement	41
Exception	
Class Exception Représente une exception	45
Frequency	
Class Frequency Représente une fréquence de répétition	46
Container< T >::Iterator	
Class Iterator Représente l'itérateur de la classe Container	49
Operation	
Class Operation Représente une opération	50
Parameters	
Class Parameters Représente le gestionnaire des paramètres de l'application	53
Ui::Principale	59
Principale	60
QCustomFrame	
Class QCustomFrame Représente un QFrame personnalisé permettant l'affichage d'un graphique	61
qt_meta_stringdata_Parameters_t	64
qt_meta_stringdata_Principale_t	64
qt_meta_stringdata_QCustomFrame_t	64
Ui_Principale	65

Chapitre 5

Index des fichiers

5.1 Liste des fichiers

Liste de tous les fichiers documentés avec une brève description :

C:/Users/Jérémy/Desktop/projets/Budgettiseur/Budgettiseur/ principale.h	??
C:/Users/Jérémy/Desktop/projets/Budgettiseur/Budgettiseur/header/ Calendar.h	69
C:/Users/Jérémy/Desktop/projets/Budgettiseur/Budgettiseur/header/ Container.h	72
C:/Users/Jérémy/Desktop/projets/Budgettiseur/Budgettiseur/header/ customClass.h	??
C:/Users/Jérémy/Desktop/projets/Budgettiseur/Budgettiseur/header/ Event.h	72
C:/Users/Jérémy/Desktop/projets/Budgettiseur/Budgettiseur/header/ Exception.h	74
C:/Users/Jérémy/Desktop/projets/Budgettiseur/Budgettiseur/header/ Parameters.h	75
C:/Users/Jérémy/Desktop/projets/Budgettiseur/build-Budgettiseur-Desktop_Qt_5_15_0_MinGW_32_bit- Debug/ ui_principale.h	??
C:/Users/Jérémy/Desktop/projets/Budgettiseur/build-Budgettiseur-Desktop_Qt_5_15_0_MinGW_32_bit- Debug/debug/ moc_predefs.h	??
C:/Users/Jérémy/Desktop/projets/Budgettiseur/build-Budgettiseur-Desktop_Qt_5_15_0_MinGW_64_bit- Debug/ ui_principale.h	??
C:/Users/Jérémy/Desktop/projets/Budgettiseur/build-Budgettiseur-Desktop_Qt_5_15_0_MinGW_64_bit- Debug/debug/ moc_predefs.h	??

Chapitre 6

Documentation des espaces de nommage

6.1 Référence de l'espace de nommage Ui

Classes

— class [Principale](#)

6.1.1 Description détaillée

Copyright

Lukos Budgeting Application copyright © 2020 Jérémy Godde

License :

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <https://www.gnu.org/licenses/>.

Chapitre 7

Documentation des classes

7.1 Référence de la classe Budget

class `Budget` Represente un budget

```
#include <Calendar.h>
```

Est dérivée de `Container< Operation * >`.

Fonctions membres publiques

- double `getBalance` () const
- double `getForecastBalance` () const
- double `getDiffBtwBalances` () const
- void `setEdit` (double _forecast_balance)
modifie le solde prévisionnel
- void `load` (QXmlStreamReader *_xml)
chargement au travers du flux xml
- void `save` (QXmlStreamWriter *_xml) const
sauvegarde au travers du flux xml
- void `show` () override
affiche les informations du budget et de ses opérations
- bool `operator<` (`Budget` const &_budget) const
- bool `operator<` (`Budget` const *_budget) const
- virtual `~Budget` ()=default
Destructeur par défaut.

Fonctions membres protégées

- `Budget` ()
constructeur sans argument pour être contenu par Containeur
- `Budget` (double _forecast_balance, `Date` *const _date)
constructeur
- void `addOperation` (QString const _subject, QString const _reference, `Date` *const _date, double _value, QString const _description)
ajoute une opération au budget

Attributs protégés

- double [balance](#)
solde réel du budget
- double [forecast_balance](#)
solde prévisionnel du budget
- [Date](#) * [date_fin](#)
date de fin du budget

Amis

- class **BudgetCategory**

7.1.1 Description détaillée

class [Budget](#) Represente un budget

Voir également

[Container](#)
[Operation](#)
[BudgetCategory](#)

7.1.2 Documentation des fonctions membres

7.1.2.1 addOperation()

```
void Budget::addOperation (
    QString const _subject,
    QString const _reference,
    Date *const _date,
    double _value,
    QString const _description ) [protected]
```

ajoute une opération au budget

Paramètres

_subject	QString const : intitulé de l'opération
_reference	QString const : référence de l'opération
_date	Date* const : date de l'opération
_value	double : montant de l'opération
_description	QString const : description de l'opération

Voir également

[Operation](#)

[Date](#)

7.1.2.2 getBalance()

```
double Budget::getBalance ( ) const
```

Renvoie

le solde réel du budget

7.1.2.3 getDiffBtwnBalances()

```
double Budget::getDiffBtwnBalances ( ) const
```

Renvoie

la différence entre les soldes prévisionnel et réel du budget

7.1.2.4 getForecastBalance()

```
double Budget::getForecastBalance ( ) const
```

Renvoie

le solde prévisionnel du budget

7.1.2.5 load()

```
void Budget::load (
    QDomStreamReader * _xml )
```

chargement au travers du flux xml

Paramètres

<code>_xml</code>	<code>QDomStreamWriter* : flux xml</code>
-------------------	---

Voir également

[Operation](#)

7.1.2.6 operator<() [1/2]

```
bool Budget::operator< (
    Budget const & _budget ) const
```

Paramètres

<code>_budget</code>	<code>Budget</code> const& : budget avec laquelle comparer
----------------------	--

Voir également

[Date](#)

Renvoie

vrai si la date du budget est antérieure à la date du budget passé en argument

7.1.2.7 operator<() [2/2]

```
bool Budget::operator< (
    Budget const * _budget ) const
```

Paramètres

<code>_budget</code>	<code>Budget</code> const* : budget avec laquelle comparer
----------------------	--

Voir également

[Date](#)

Renvoie

vrai si la date du budget est antérieure à la date du budget passé en argument

7.1.2.8 save()

```
void Budget::save (
    QDomStreamWriter * _xml ) const
```

sauvegarde au travers du flux xml

Paramètres

<code>_xml</code>	<code>QXmlStreamWriter* : flux xml</code>
-------------------	---

Voir également

[Operation](#)

7.1.2.9 setEdit()

```
void Budget::setEdit (
    double _forecast_balance )
```

modifie le solde prévisionnel

Paramètres

<code>_forecast_balance</code>	<code>double : nouveau solde prévisionnel</code>
--------------------------------	--

7.1.2.10 show()

```
void Budget::show ( ) [override], [virtual]
```

affiche les informations du budget et de ses opérations

Voir également

[Container](#)

Implémente [Container< Operation * >](#).

La documentation de cette classe a été générée à partir du fichier suivant :

- `C:/Users/Jérémy/Desktop/projets/Budgettiseur/Budgettiseur/header/Calendar.h`
- `C:/Users/Jérémy/Desktop/projets/Budgettiseur/Budgettiseur/source/Calendar.cpp`

7.2 Référence de la classe BudgetCategory

class [BudgetCategory](#) Represente une catégorie budgétaire contenant plusieurs budgets

```
#include <Calendar.h>
```

Est dérivée de [Event](#), et [Container< Budget * >](#).

Types publics

- enum `Type` {
undefined = 0, **undefined** = 0, **credit** = 1, **credit** = 1,
debit = 2, **debit** = 2 }
ensemble des types de budget

Fonctions membres publiques

- `Type` `getType` () const
- `QRgb` `getColor` () const
- void `setSubject` (QString const _subject)
modifie l'intitulé
- void `setDate` (Date *const _date)
- void `setFrequency` (Frequency *const _frequency)
- void `setOccurence` (unsigned _occurence)
modifie l'occurence
- void `setType` (Type _type)
modifie le type
- void `setColor` (QRgb _color)
modifie la couleur
- void `addOperation` (QString const _subject, QString const _reference, Date *const _date, double _value, QString const _description)
ajoute une opération au budget
- void `delOperation` (Operation *_element)
supprime l'opération de la catégorie ou ne fais rien si l'opération ne fait pas partie de la catégorie
- void `load` (QXmlStreamReader *_xml)
chargement au travers du flux xml
- void `save` () override
effectue une sauvegarde des données de la catégorie
- void `save` (QXmlStreamWriter *_xml) const
sauvegarde au travers du flux xml
- void `show` () override
affiche les informations de la catégorie et de ses budgets
- virtual `~BudgetCategory` ()=default
Destructeur par défaut.

Fonctions membres protégées

- `BudgetCategory` ()
constructeur sans argument pour être contenu par Conteneur
- `BudgetCategory` (QString const _subject, Date *const _date, Frequency *const _frequency, unsigned _occurence, Type _type)
constructeur
- void `draw` ()
dessine une représentation graphique de la catégorie de budget : montants en fonction des budgets
- double `getSumBalances` ()

Attributs protégés

- `QRgb` `color` = `qRgb(0,0,0)`
couleur associée à la catégorie
- `Type` `type` = `undefined`
type de la catégorie

Amis

- class `Calendar`

7.2.1 Description détaillée

class [BudgetCategory](#) Represente une catégorie budgétaire contenant plusieurs budgets

Voir également

[Event](#)
[Container](#)
[Budget](#)
[Calendar](#)

7.2.2 Documentation des fonctions membres

7.2.2.1 addOperation()

```
void BudgetCategory::addOperation (
    QString const _subject,
    QString const _reference,
    Date *const _date,
    double _value,
    QString const _description )
```

ajoute une opération au budget

Paramètres

<i>_subject</i>	QString const : intitulé de l'opération
<i>_reference</i>	QString const : référence de l'opération
<i>_date</i>	Date* const : date de l'opération
<i>_value</i>	double : montant de l'opération
<i>_description</i>	QString const : description de l'opération

Voir également

[Operation](#)
[Date](#)
[Budget](#)

7.2.2.2 delOperation()

```
void BudgetCategory::delOperation (
    Operation * _element )
```

supprime l'opération de la catégorie ou ne fais rien si l'opération ne fait pas partie de la catégorie

Paramètres

<code>_element</code>	Operation* : opération à supprimer
-----------------------	------------------------------------

7.2.2.3 getColor()

```
QRgb BudgetCategory::getColor ( ) const
```

Renvoie

la couleur associée à la catégorie

7.2.2.4 getSumBalances()

```
double BudgetCategory::getSumBalances ( ) [protected]
```

Renvoie

la somme de tous les soldes de la catégorie

7.2.2.5 getType()

```
BudgetCategory::Type BudgetCategory::getType ( ) const
```

Renvoie

le type de la catégorie

7.2.2.6 load()

```
void BudgetCategory::load (
    QDomStreamReader * _xml )
```

chargement au travers du flux xml

Paramètres

<code>_xml</code>	QXmlStreamWriter* : flux xml
-------------------	------------------------------

Voir également

[Budget](#)

7.2.2.7 save() [1/2]

```
void BudgetCategory::save ( ) [override], [virtual]
```

effectue une sauvegarde des données de la catégorie

Voir également

[Event](#)

Implémente [Event](#).

7.2.2.8 save() [2/2]

```
void BudgetCategory::save (
    QDomStreamWriter * _xml ) const
```

sauvegarde au travers du flux xml

Paramètres

<code>_xml</code>	QDomStreamWriter* : flux xml
-------------------	------------------------------

Voir également

[Budget](#)

7.2.2.9 setColor()

```
void BudgetCategory::setColor (
    QColor _color )
```

modifie la couleur

Paramètres

<code>_color</code>	QColor : nouvelle couleur
---------------------	---------------------------

7.2.2.10 setOccurence()

```
void BudgetCategory::setOccurence (
    unsigned _occurence )
```

modifie l'occurence

Paramètres

<code>_occurence</code>	unsigned : nouveau occurence
-------------------------	------------------------------

7.2.2.11 setSubject()

```
void BudgetCategory::setSubject (
    QString const _subject )
```

modifie l'intitulé

Paramètres

<code>_subject</code>	QString const : nouvel intitulé
-----------------------	---------------------------------

7.2.2.12 setType()

```
void BudgetCategory::setType (
    BudgetCategory::Type _type )
```

modifie le type

Paramètres

<code>_type</code>	<code>BudgetCategory::Type</code> : nouveau type
--------------------	--

7.2.2.13 show()

```
void BudgetCategory::show ( ) [override], [virtual]
```

affiche les informations de la catégorie et de ses budgets

Voir également

[Container](#)

Implémente [Container< Budget * >](#).

La documentation de cette classe a été générée à partir du fichier suivant :

- C:/Users/Jérémy/Desktop/projets/Budgettiseur/Budgettiseur/header/[Calendar.h](#)
- C:/Users/Jérémy/Desktop/projets/Budgettiseur/Budgettiseur/source/Calendar.cpp

7.3 Référence de la classe Calendar

class [Calendar](#) Représente un calendrier budgétaire contenant plusieurs catégories budgétaires

```
#include <Calendar.h>
```

Est dérivée de [Container< BudgetCategory * >](#).

Fonctions membres publiques

- void [addBudgetCategory](#) (QString const _subject, [Date](#) *const _date, [Frequency](#) *const _frequency, unsigned _occurrence, [BudgetCategory::Type](#) _type, QColor _color)
ajoute une catégorie au calendrier
- double [getBalance](#) () const
- void [addBalance](#) (const double _value)
modifie le solde réel du calendrier
- void [remove](#) ([BudgetCategory](#) *_element) override
supprime l'élément indiqué
- void [load](#) (QXmlStreamReader *_xml)
chargement au travers du flux xml
- void [save](#) (QXmlStreamWriter *_xml) const
sauvegarde au travers du flux xml
- [BudgetCategory](#) * [operator>>](#) (QString _subject)
- void [show](#) () override
affiche les informations du calendrier et de ses catégories budgétaires
- virtual [~Calendar](#) ()=default
Destructeur par défaut.

Fonctions membres publiques statiques

- static [Calendar](#) & [getInstance](#) ()

Fonctions membres protégées

- [Calendar](#) ()
constructeur

Attributs protégés

- double [balance](#) = 0
solde réel du calendrier

Attributs protégés statiques

- static [Calendar](#) * [calendar](#) = new [Calendar](#)()
instance unique du calendrier

7.3.1 Description détaillée

class [Calendar](#) Represente un calendrier budgétaire contenant plusieurs catégories budgétaires

Voir également

[Container](#)

[BudgetCategory](#)

7.3.2 Documentation des fonctions membres

7.3.2.1 addBudgetCategory()

```
void Calendar::addBudgetCategory (
    QString const _subject,
    Date *const _date,
    Frequency *const _frequency,
    unsigned _occurence,
    BudgetCategory::Type _type,
    QRgb _color )
```

ajoute une catégorie au calendrier

Paramètres

<code>_subject</code>	QString const : intitulé de la catégorie
<code>_date</code>	Date* const : date de départ de la catégorie
<code>_frequency</code>	Frequency* const : fréquence de la catégorie
<code>_occurence</code>	unsigned : occurrence de la catégorie
<code>_type</code>	BudgetCategory::Type : type de la catégorie
<code>_color</code>	QRgb : couleur de la catégorie

Voir également

[Date](#)

[Frequency](#)

[BudgetCategory](#)

7.3.2.2 getBalance()

```
double Calendar::getBalance ( ) const
```

Renvoie

le solde réel du calendrier

7.3.2.3 getInstance()

```
Calendar & Calendar::getInstance ( ) [static]
```

Renvoie

l'instance unique du calendrier

7.3.2.4 load()

```
void Calendar::load (
    QDomStreamReader * _xml )
```

chargement au travers du flux xml

Paramètres

<code>_xml</code>	QDomStreamWriter* : flux xml
-------------------	------------------------------

Voir également

[BudgetCategory](#)

7.3.2.5 operator>>()

```
BudgetCategory * Calendar::operator>> (
    QString _subject )
```

Paramètres

<code>_subject</code>	QString : l'intitulé de la catégorie à rechercher
-----------------------	---

Renvoie

la catégorie correspondant à l'intitulé ou nullptr si aucune catégorie ne correspond à l'intitulé

7.3.2.6 remove()

```
void Calendar::remove (
    BudgetCategory * _element ) [override], [virtual]
```

supprime l'élément indiqué

Paramètres

<code>_element</code>	T : élément à supprimer
-----------------------	-------------------------

Réimplémentée à partir de [Container< BudgetCategory * >](#).

7.3.2.7 save()

```
void Calendar::save (
    QDomStreamWriter * _xml ) const
```

sauvegarde au travers du flux xml

Paramètres

<code>_xml</code>	QDomStreamWriter* : flux xml
-------------------	------------------------------

Voir également

[BudgetCategory](#)

7.3.2.8 show()

```
void Calendar::show ( ) [override], [virtual]
```

affiche les informations du calendrier et de ses catégories budgétaires

Voir également

[Container](#)

Implémente [Container< BudgetCategory * >](#).

7.3.3 Documentation des données membres

7.3.3.1 calendar

```
Calendar * Calendar::calendar = new Calendar() [static], [protected]
```

instance unique du calendrier

Copyright

Lukos Budgeting Application copyright © 2020 Jérémy Godde

License :

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <https://www.gnu.org/licenses/>.

La documentation de cette classe a été générée à partir du fichier suivant :

- C:/Users/Jérémy/Desktop/projets/Budgettiseur/Budgettiseur/header/Calendar.h
- C:/Users/Jérémy/Desktop/projets/Budgettiseur/Budgettiseur/source/Calendar.cpp

7.4 Référence du modèle de la classe Container< T >

class Container<T> Représente un contenant d'objets T

```
#include <Container.h>
```

Classes

- class `Iterator`
class `Iterator` Représente l'itérateur de la classe `Container`

Fonctions membres publiques

- `Container ()`
constructeur
- `Iterator begin ()`
- `unsigned getSize () const`
- `void sort ()`
trie le tableau
- `virtual void remove (unsigned _position)`
supprime l'élément à la position indiquée
- `virtual void remove (T _element)`
supprime l'élément indiqué
- `void insert (T _element)`
insère l'élément dans le tableau
- `T operator[] (unsigned _position)`
- `T operator[] (unsigned _position) const`
- `virtual void show ()=0`
affiche les informations du conteneur et de ses éléments

Attributs protégés

- T * `array_elements`
tableau d'éléments
- unsigned `size`
taille du tableau

7.4.1 Description détaillée

```
template<typename T>
class Container< T >
```

class Container<T> Représente un contenant d'objets T

Remarques

nécessite pour T un constructeur sans argument

7.4.2 Documentation des constructeurs et destructeur

7.4.2.1 Container()

```
template<typename T >
Container< T >::Container [explicit]
```

constructeur

Copyright

Lukos Budgeting Application copyright © 2020 Jérémy Godde

License :

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <https://www.gnu.org/licenses/>.

7.4.3 Documentation des fonctions membres

7.4.3.1 `begin()`

```
template<typename T >
Container< T >::Iterator Container< T >::begin
```

Renvoie

un `Iterator` positionné au début

Voir également

`Iterator`

7.4.3.2 `getSize()`

```
template<typename T >
unsigned Container< T >::getSize
```

Renvoie

la taille du tableau

7.4.3.3 `insert()`

```
template<typename T >
void Container< T >::insert (
    T _element )
```

insère l'élément dans le tableau

Paramètres

<code>_element</code>	T : élément à insérer
-----------------------	-----------------------

7.4.3.4 `operator[]()` [1/2]

```
template<typename T >
T Container< T >::operator[] (
    unsigned _position )
```

Paramètres

<code>_position</code>	unsigned : position de l'élément à retourner
------------------------	--

Renvoie

l'élément à la position indiquée

7.4.3.5 operator[]() [2/2]

```
template<typename T >
T Container< T >::operator[] (
    unsigned _position ) const
```

Paramètres

<code>_position</code>	unsigned : position de l'élément à retourner
------------------------	--

Renvoie

l'élément à la position indiquée

7.4.3.6 remove() [1/2]

```
template<typename T >
void Container< T >::remove (
    T _element ) [virtual]
```

supprime l'élément indiqué

Paramètres

<code>_element</code>	T : élément à supprimer
-----------------------	-------------------------

Réimplémentée dans [Calendar](#).

7.4.3.7 remove() [2/2]

```
template<typename T >
void Container< T >::remove (
    unsigned _position ) [virtual]
```

supprime l'élément à la position indiquée

Paramètres

<code>_position</code>	unsigned : position de l'élément à supprimer
------------------------	--

La documentation de cette classe a été générée à partir du fichier suivant :

- C:/Users/Jérémy/Desktop/projets/Budgettiseur/Budgettiseur/header/Container.h
- C:/Users/Jérémy/Desktop/projets/Budgettiseur/Budgettiseur/source/Container.cpp

7.5 Référence de la classe Date

class [Date](#) Représente une date.

```
#include <Event.h>
```

Types publics

- enum [Day](#) {
`d_1 = 1, d_1 = 1, d_2 = 2, d_2 = 2,`
`d_3 = 3, d_3 = 3, d_4 = 4, d_4 = 4,`
`d_5 = 5, d_5 = 5, d_6 = 6, d_6 = 6,`
`d_7 = 7, d_7 = 7, d_8 = 8, d_8 = 8,`
`d_9 = 9, d_9 = 9, d_10 = 10, d_10 = 10,`
`d_11 = 11, d_11 = 11, d_12 = 12, d_12 = 12,`
`d_13 = 13, d_13 = 13, d_14 = 14, d_14 = 14,`
`d_15 = 15, d_15 = 15, d_16 = 16, d_16 = 16,`
`d_17 = 17, d_17 = 17, d_18 = 18, d_18 = 18,`
`d_19 = 19, d_19 = 19, d_20 = 20, d_20 = 20,`
`d_21 = 21, d_21 = 21, d_22 = 22, d_22 = 22,`
`d_23 = 23, d_23 = 23, d_24 = 24, d_24 = 24,`
`d_25 = 25, d_25 = 25, d_26 = 26, d_26 = 26,`
`d_27 = 27, d_27 = 27, d_28 = 28, d_28 = 28,`
`d_29 = 29, d_29 = 29, d_30 = 30, d_30 = 30,`
`d_31 = 31, d_31 = 31 }`
ensemble des jours
- enum [Month](#) {
`january = 1, january = 1, february = 2, february = 2,`
`march = 3, march = 3, april = 4, april = 4,`
`may = 5, may = 5, june = 6, june = 6,`
`july = 7, july = 7, august = 8, august = 8,`
`september = 9, september = 9, october = 10, october = 10,`
`november = 11, november = 11, december = 12, december = 12 }`
ensemble des mois
- typedef unsigned [Year](#)
représente l'année

Fonctions membres publiques

- [Date](#) ([Day](#) _day, [Month](#) _month, [Year](#) _year)
constructeur
- unsigned [getDay](#) () const
- unsigned [getMonth](#) () const
- unsigned [getYear](#) () const
- QString const [toString](#) (QString const _format="dd mmmm yyyy") const
- QDate const [toQDate](#) () const
- bool [operator<](#) ([Date](#) const &_date) const
- bool [operator<](#) ([Date](#) const *_date) const
- [Date](#) [operator+](#) ([Frequency](#) const &_frequency) const
- [Date](#) [operator-](#) ([Frequency](#) const &_frequency) const
- [Date](#) & [operator=](#) ([Date](#) const &_date)=default
opérateur de recopie par défaut

Fonctions membres publiques statiques

- static [Date](#) const [getToday](#) ()
- static [Date](#) * [fromString](#) (QString _text)

Attributs protégés

- [Day](#) *day*
jour
- [Month](#) *month*
mois
- [Year](#) *year*
année

7.5.1 Description détaillée

class [Date](#) Représente une date.

7.5.2 Documentation des fonctions membres

7.5.2.1 fromString()

```
Date * Date::fromString (  
    QString _text ) [static]
```

Paramètres

<code>_text</code>	QString : texte à convertir
--------------------	-----------------------------

Renvoie

un pointeur sur un objet date correspondant au texte

7.5.2.2 getDay()

```
unsigned Date::getDay ( ) const
```

Renvoie

le jour de la date

7.5.2.3 getMonth()

```
unsigned Date::getMonth ( ) const
```

Renvoie

le mois de la date

7.5.2.4 getToday()

```
Date const Date::getToday ( ) [static]
```

Renvoie

la date du jour

7.5.2.5 getYear()

```
unsigned Date::getYear ( ) const
```

Renvoie

l'année de la date

7.5.2.6 operator+()

```
Date Date::operator+ (
    Frequency const & _frequency ) const
```

Paramètres

<code>_frequency</code>	<code>Frequency</code> const* : fréquence à ajouter
-------------------------	---

Voir également

[Frequency](#)

Renvoie

la date correspondant à la somme de la date et de la fréquence

7.5.2.7 operator-()

```
Date Date::operator- (
    Frequency const & _frequency ) const
```

Paramètres

<code>_frequency</code>	<code>Frequency</code> const* : fréquence à soustraire
-------------------------	--

Voir également

[Frequency](#)

Renvoie

la date correspondant à la différence de la date et de la fréquence

7.5.2.8 operator<() [1/2]

```
bool Date::operator< (
    Date const & _date ) const
```

Paramètres

<code>_date</code>	<code>Date</code> const& : date avec laquelle comparer
--------------------	--

Renvoie

vrai si la date est antérieure à la date passée en argument

7.5.2.9 operator<() [2/2]

```
bool Date::operator< (
    Date const * _date ) const
```

Paramètres

<code>_date</code>	<code>Date</code> const* : date avec laquelle comparer
--------------------	--

Renvoie

vrai si la date est antérieure à la date passée en argument

7.5.2.10 toQDate()

```
QDate const Date::toQDate ( ) const
```

Renvoie

l'objet QDate correspondant à la date

7.5.2.11 toString()

```
QString const Date::toString (
    QString const _format = "dd mmmm yyyy" ) const
```

Paramètres

<code>_format</code>	QString const : format de la date
----------------------	-----------------------------------

Renvoie

la date au format indiqué

La documentation de cette classe a été générée à partir du fichier suivant :

- C:/Users/Jérémy/Desktop/projets/Budgettiseur/Budgettiseur/header/Event.h
- C:/Users/Jérémy/Desktop/projets/Budgettiseur/Budgettiseur/source/Event.cpp

7.6 Référence de la classe Event

class [Event](#) Représente un événement.

```
#include <Event.h>
```

Dérivée par [BudgetCategory](#).

Fonctions membres publiques

- [Date](#) const [getDate](#) (unsigned _occurence=0) const
- [Frequency](#) const & [getFrequency](#) () const
- unsigned [getOccurence](#) () const
- QString const [getSubject](#) () const
- bool [operator<](#) ([Event](#) const * _event) const
- bool [operator<](#) ([Event](#) const & _event) const
- bool [operator>](#) ([Event](#) const * _event) const
- bool [operator>](#) ([Event](#) const & _event) const
- [Event](#) & [operator=](#) ([Event](#) const & _event)=default
opérateur de recopie par défaut
- virtual void [save](#) ()=0
effectue une sauvegarde des données de l'événement

Fonctions membres protégées

- [Event](#) (QString const _subject, [Date](#) *const _date, [Frequency](#) *const _frequency, unsigned _occurence)
constructeur

Attributs protégés

- [Date](#) * [date_start](#)
date de début de l'événement
- [Frequency](#) * [frequency](#)
fréquence de l'événement
- unsigned [occurence](#)
occurence de l'événement, si égale à 0 alors le nombre d'occurrences est indéterminé
- QString [subject](#)
sujet de l'événement

7.6.1 Description détaillée

class [Event](#) Représente un événement.

7.6.2 Documentation des fonctions membres

7.6.2.1 getDate()

```
Date const Event::getDate (
    unsigned \_occurence = 0 ) const
```

Paramètres

_occurence	unsigned : occurrence de l'événement pour laquelle retourner la date
----------------------------	--

Voir également

[Date](#)

Renvoie

la date correspondant à l'occurrence choisie de l'événement

7.6.2.2 getFrequency()

```
Frequency const & Event::getFrequency ( ) const
```

Voir également

[Frequency](#)

Renvoie

la fréquence de l'événement

7.6.2.3 getOccurence()

```
unsigned Event::getOccurence ( ) const
```

Renvoie

le nombre d'occurence

7.6.2.4 getSubject()

```
QString const Event::getSubject ( ) const
```

Renvoie

le sujet de l'événement

7.6.2.5 operator<() [1/2]

```
bool Event::operator< (
    Event const & _event ) const
```

Paramètres

<code>_event</code>	<code>Event</code> const& : événement avec lequel comparer
---------------------	--

Voir également

[Date](#)

Renvoie

vrai si la date de l'événement est antérieure à la date de l'événement passé en argument

7.6.2.6 operator<() [2/2]

```
bool Event::operator< (
    Event const * _event ) const
```

Paramètres

<code>_event</code>	<code>Event</code> const* : événement avec lequel comparer
---------------------	--

Voir également

[Date](#)

Renvoie

vrai si la date de l'événement est antérieure à la date de l'événement passé en argument

7.6.2.7 `operator>()` [1/2]

```
bool Event::operator> (
    Event const & _event ) const
```

Paramètres

<code>_event</code>	<code>Event</code> const& : événement avec lequel comparer
---------------------	--

Voir également

[Date](#)

Renvoie

vrai si la date de l'événement est postérieure à la date de l'événement passé en argument

7.6.2.8 `operator>()` [2/2]

```
bool Event::operator> (
    Event const * _event ) const
```

Paramètres

<code>_event</code>	<code>Event</code> const* : événement avec lequel comparer
---------------------	--

Voir également

[Date](#)

Renvoie

vrai si la date de l'événement est postérieure à la date de l'événement passé en argument

La documentation de cette classe a été générée à partir du fichier suivant :

- C:/Users/Jérémy/Desktop/projets/Budgettiseur/Budgettiseur/header/[Event.h](#)
- C:/Users/Jérémy/Desktop/projets/Budgettiseur/Budgettiseur/source/Event.cpp

7.7 Référence de la classe Exception

class [Exception](#) Represente une exception

```
#include <Exception.h>
```

Fonctions membres publiques

- [Exception](#) (QString _message)
constructeur
- QString [getString](#) () const
- std::string [getStdString](#) () const

Attributs protégés

- QString [message](#)
message explicatif de la cause de l'exception

7.7.1 Description détaillée

class [Exception](#) Represente une exception

7.7.2 Documentation des fonctions membres

7.7.2.1 getStdString()

```
std::string Exception::getStdString ( ) const [inline]
```

Renvoie

le message comme un std::string

7.7.2.2 getString()

```
QString Exception::getString ( ) const [inline]
```

Renvoie

le message comme un QString

La documentation de cette classe a été générée à partir du fichier suivant :

- C:/Users/Jérémy/Desktop/projets/Budgettiseur/Budgettiseur/header/[Exception.h](#)

7.8 Référence de la classe Frequency

class [Frequency](#) Représente une fréquence de répétition.

```
#include <Event.h>
```

Types publics

- enum **Rythm** { WEEKLY, MONTHLY, YEARLY }

Fonctions membres publiques

- [Frequency](#) (Rythm _rythm, unsigned _frequency)
constructeur
- Rythm [getRythm](#) () const
- unsigned [getFrequency](#) () const
- unsigned [getNbWeek](#) () const
- unsigned [getNbMonth](#) () const
- unsigned [getNbYear](#) () const
- QString const [toString](#) () const
- [Frequency operator*](#) (unsigned const &_factor) const

Fonctions membres publiques statiques

- static [Frequency](#) * [fromString](#) (QString _text)

Attributs protégés

- unsigned [frequency](#)
valeur de fréquence
- Rythm [rythm](#)
rythme de la fréquence

7.8.1 Description détaillée

class [Frequency](#) Représente une fréquence de répétition.

7.8.2 Documentation des constructeurs et destructeur

7.8.2.1 Frequency()

```
Frequency::Frequency (
    Rythm _rythm,
    unsigned _frequency ) [explicit]
```

constructeur

Copyright

Lukos Budgeting Application copyright © 2020 Jérémy Godde

License :

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <https://www.gnu.org/licenses/>.

7.8.3 Documentation des fonctions membres

7.8.3.1 fromString()

```
Frequency * Frequency::fromString (
    QString _text ) [static]
```

Paramètres

<code>_text</code>	QString : texte à convertir
--------------------	-----------------------------

Renvoie

un pointeur sur un objet fréquence correspondant au texte

7.8.3.2 getFrequency()

```
unsigned Frequency::getFrequency ( ) const
```

Renvoie

la fréquence

7.8.3.3 getNbMonth()

```
unsigned Frequency::getNbMonth ( ) const
```

Renvoie

la part de la fréquence correspondant aux mois

7.8.3.4 getNbWeek()

```
unsigned Frequency::getNbWeek ( ) const
```

Renvoie

la part de la fréquence correspondant aux semaines

7.8.3.5 getNbYear()

```
unsigned Frequency::getNbYear ( ) const
```

Renvoie

la part de la fréquence correspondant aux années

7.8.3.6 getRythm()

```
Frequency::Rythm Frequency::getRythm ( ) const
```

Renvoie

le rythme de la fréquence

7.8.3.7 operator*()

```
Frequency Frequency::operator* (
    unsigned const & _factor ) const
```

Paramètres

<code>_factor</code>	unsigned const& : occurrence choisie
----------------------	--------------------------------------

Renvoie

la fréquence de l'occurrence choisie

7.8.3.8 toString()

```
QString const Frequency::toString ( ) const
```

Renvoie

la fréquence au format dd mm yyyy

La documentation de cette classe a été générée à partir du fichier suivant :

- C:/Users/Jérémy/Desktop/projets/Budgettiseur/Budgettiseur/header/Event.h
- C:/Users/Jérémy/Desktop/projets/Budgettiseur/Budgettiseur/source/Event.cpp

7.9 Référence de la classe `Container< T >::Iterator`

class `Iterator` Représente l'itérateur de la classe `Container`

```
#include <Container.h>
```

Fonctions membres publiques

- `Iterator` (`Container< T > &_container`, unsigned `_position`)
constructeur
- bool `isTerminated` () const
- `Iterator & operator++` (int)
incrémente l'itérateur avant de le retourner
- T `operator*` ()
- `Iterator & operator=` (`Iterator` const &`_iterator`)=default
opérateur de copie par défaut

Attributs protégés

- unsigned `position`
position actuelle de l'itérateur
- `Container< T > & container`
position actuelle de l'itérateur
- bool `termine`
marqueur indiquant si l'itérateur est arrivé à la fin

7.9.1 Description détaillée

```
template<typename T>
class Container< T >::Iterator
```

class `Iterator` Représente l'itérateur de la classe `Container`

Voir également

`Container`

7.9.2 Documentation des fonctions membres

7.9.2.1 isTerminated()

```
template<typename T >
bool Container< T >::Iterator::isTerminated
```

Renvoie

vrai si l'itérateur a atteint une extrémité

7.9.2.2 operator*()

```
template<typename T >
T Container< T >::Iterator::operator*
```

Renvoie

l'élément sur lequel pointe l'itérateur

7.9.2.3 operator++()

```
template<typename T >
Container< T >::Iterator & Container< T >::Iterator::operator++ (
    int )
```

incrémente l'itérateur avant de le retourner

Renvoie

retourne l'itérateur

La documentation de cette classe a été générée à partir du fichier suivant :

- C:/Users/Jérémy/Desktop/projets/Budgettiseur/Budgettiseur/header/[Container.h](#)
- C:/Users/Jérémy/Desktop/projets/Budgettiseur/Budgettiseur/source/Container.cpp

7.10 Référence de la classe Operation

class [Operation](#) Represente une opération

```
#include <Calendar.h>
```

Fonctions membres publiques

- QString const [getSubject](#) () const
- QString const [getReference](#) () const
- double [getValue](#) () const
- [Date](#) const & [getDate](#) () const
- QString const [getDescription](#) () const
- void **setEdit** (QString const _subject, QString const _reference, [Date](#) *const _date, double _value, QString const _description)
- void **load** (QXmlStreamReader * _xml)
chargement au travers du flux xml
- void **save** (QXmlStreamWriter * _xml) const
sauvegarde au travers du flux xml
- bool **operator<** ([Operation](#) const & _operation) const
- bool **operator<** ([Operation](#) const * _operation) const

Fonctions membres protégées

- [Operation](#) ()
constructeur sans argument pour être contenu par Containeur
- [Operation](#) (QString const _subject, QString const _reference, [Date](#) *const _date, double _value, QString const _description)
constructeur

Attributs protégés

- QString [subject](#)
intitulé de l'opération
- QString [reference](#)
référence de l'opération
- double [value](#)
montant de l'opération
- [Date](#) * [date](#)
date de l'opération
- QString [description](#)
description de l'opération

Amis

- class **Budget**

7.10.1 Description détaillée

class [Operation](#) Represente une opération

Voir également

[Budget](#)

7.10.2 Documentation des fonctions membres

7.10.2.1 getDate()

```
Date const & Operation::getDate ( ) const
```

Voir également

[Date](#)

Renvoie

la date de l'opération

7.10.2.2 getDescription()

```
QString const Operation::getDescription ( ) const
```

Renvoie

la description de l'opération

7.10.2.3 getReference()

```
QString const Operation::getReference ( ) const
```

Renvoie

la référence de l'opération

7.10.2.4 getSubject()

```
QString const Operation::getSubject ( ) const
```

Renvoie

l'intitulé de l'opération

7.10.2.5 getValue()

```
double Operation::getValue ( ) const
```

Renvoie

le montant de l'opération

7.10.2.6 load()

```
void Operation::load (
    QDomStreamReader * _xml )
```

chargement au travers du flux xml

Paramètres

<code>_xml</code>	<code>QXmlStreamWriter* : flux xml</code>
-------------------	---

7.10.2.7 operator<()

```
bool Operation::operator< (
    Operation const & _operation ) const
```

Paramètres

<code>_operation</code>	<code>Operation const& : opération avec laquelle comparer</code>
-------------------------	--

Voir également

[Date](#)

Renvoie

vrai si la date de l'opération est antérieure à la date de l'opération passée en argument

7.10.2.8 save()

```
void Operation::save (
    QXmlStreamWriter * _xml ) const
```

sauvegarde au travers du flux xml

Paramètres

<code>_xml</code>	<code>QXmlStreamWriter* : flux xml</code>
-------------------	---

La documentation de cette classe a été générée à partir du fichier suivant :

- C:/Users/Jérémy/Desktop/projets/Budgettiseur/Budgettiseur/header/[Calendar.h](#)
- C:/Users/Jérémy/Desktop/projets/Budgettiseur/Budgettiseur/source/Calendar.cpp

7.11 Référence de la classe Parameters

class [Parameters](#) Represente le gestionnaire des paramètres de l'application

```
#include <Parameters.h>
```

Est dérivée de QObject.

Connecteurs publics

- void [changeDevise](#) (QString const _devise)
modifie la devise
- void [changeDeviseVisible](#) (bool _visible)
modifie l'état d'affichage de la devise dans les tableaux
- void [changeFormatDate](#) (QString const _format)
modifie le format d'affichage des dates
- void [changeMaxPoints](#) (unsigned _max_points)
modifie le nombre maximal de point à afficher sur le graphique
- void [changeWriteSave](#) (QString const _file)
modifie l'emplacement du fichier
- void [changeWriteRegistry](#) (QString const _dir)
modifie l'emplacement du dossier d'archivage
- void [changeWritePDF](#) (QString const _dir)
modifie l'emplacement du dossier d'export de pdf

Fonctions membres publiques

- [Parameters](#) ()
constructeur sans arguments
- [~Parameters](#) ()=default
destructeur par défaut
- bool [openHelp](#) () const
ouvre dans le navigateur par défaut la page d'aide de l'application
- QString const & [getAppName](#) () const
- QString const & [getAppLongName](#) () const
- QString const & [getAppVersion](#) () const
- QString const & [getDevise](#) () const
- bool [getDeviseVisible](#) () const
- QString const & [getDirRegistry](#) () const
- QString const & [getDirPDF](#) () const
- QString const & [getFileSave](#) () const
- QString const & [getFormatDate](#) () const
- unsigned [getGraphicsMaxPoints](#) () const
- void [save](#) ()
effectue une sauvegarde des paramètres

Attributs protégés

- QString const [cfg_save](#) = "config.xml"
emplacement du fichier de configuration
- QUrl const [url_help](#) = QUrl("https://github.com/JeremyGodde/Lukos/wiki/Aide")
url de la page d'aide
- QString [app_name](#) = "Lukos"
nom courant de l'application
- QString [app_long_name](#) = [app_name](#) + " Budgeting Application"
nom complet de l'application
- QString [app_version](#) = "Abricot (0.3)"
version actuelle de l'application
- QString [devise](#)
devise des valeurs
- bool [devise_visible](#)
affiche la devise dans les tableaux si true, sinon n'affiche pas la devise dans les tableaux
- QString [dir_registry](#)
emplacement du dossier où doivent être exportées les archives
- QString [dir_pdf](#)
emplacement du dossier où doivent être exportées les pdf
- QString [file_save](#)
emplacement du fichier de sauvegarde
- QString [format_date](#)
format d'affichage de la date
- unsigned [graphics_max_points](#)
nombre de point d'afficher sur le graphique

7.11.1 Description détaillée

class [Parameters](#) Représente le gestionnaire des paramètres de l'application

7.11.2 Documentation des constructeurs et destructeur

7.11.2.1 Parameters()

```
Parameters::Parameters ( )
```

constructeur sans arguments

Copyright

Lukos Budgeting Application copyright © 2020 Jérémy Godde

License :

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <https://www.gnu.org/licenses/>.

7.11.3 Documentation des fonctions membres

7.11.3.1 changeDevise

```
void Parameters::changeDevise (
    QString const _devise ) [inline], [slot]
```

modifie la devise

Paramètres

<code>_devise</code>	QString const : la devise
----------------------	---------------------------

7.11.3.2 changeDeviseVisible

```
void Parameters::changeDeviseVisible (
    bool _visible ) [inline], [slot]
```

modifie l'état d'affichage de la devise dans les tableaux

Paramètres

<code>_visible</code>	bool : état d'affichage de la devise dans les tableaux
-----------------------	--

7.11.3.3 `changeFormatDate`

```
void Parameters::changeFormatDate (
    QString const _format ) [inline], [slot]
```

modifie le format d'affichage des dates

Paramètres

<code>_format</code>	QString const : le format d'affichage des dates
----------------------	---

7.11.3.4 `changeMaxPoints`

```
void Parameters::changeMaxPoints (
    unsigned _max_points ) [inline], [slot]
```

modifie le nombre maximal de point à afficher sur le graphique

Paramètres

<code>_max_points</code>	unsigned : le nombre maximal de point à afficher sur le graphique
--------------------------	---

7.11.3.5 `changeWritePDF`

```
void Parameters::changeWritePDF (
    QString const _dir ) [inline], [slot]
```

modifie l'emplacement du dossier d'export de pdf

Paramètres

<code>_dir</code>	QString const : l'emplacement du dossier d'export de pdf
-------------------	--

7.11.3.6 changeWriteRegistry

```
void Parameters::changeWriteRegistry (
    QString const _dir ) [inline], [slot]
```

modifie l'emplacement du dossier d'archivage

Paramètres

<code>_dir</code>	QString const : l'emplacement du dossier d'archivage
-------------------	--

7.11.3.7 changeWriteSave

```
void Parameters::changeWriteSave (
    QString const _file ) [inline], [slot]
```

modifie l'emplacement du fichier

Paramètres

<code>_file</code>	QString const : l'emplacement du fichier de sauvegarde
--------------------	--

7.11.3.8 getAppLongName()

```
QString const & Parameters::getAppLongName ( ) const
```

Renvoie

le nom complet de l'application

7.11.3.9 getAppName()

```
QString const & Parameters::getAppName ( ) const
```

Renvoie

le nom courant de l'application

7.11.3.10 getAppVersion()

```
QString const & Parameters::getAppVersion ( ) const
```

Renvoie

la version actuelle de l'application

7.11.3.11 getDevise()

```
QString const & Parameters::getDevise ( ) const
```

Renvoie

la devise des valeurs

7.11.3.12 getDeviseVisible()

```
bool Parameters::getDeviseVisible ( ) const
```

Renvoie

si la devise doit ou non être affichée dans les tableaux

7.11.3.13 getDirPDF()

```
QString const & Parameters::getDirPDF ( ) const
```

Renvoie

l'emplacement du dossier d'export de pdf

7.11.3.14 getDirRegistry()

```
QString const & Parameters::getDirRegistry ( ) const
```

Renvoie

l'emplacement du dossier d'archivage

7.11.3.15 getFileSave()

```
QString const & Parameters::getFileSave ( ) const
```

Renvoie

l'emplacement du fichier de sauvegarde

7.11.3.16 getFormatDate()

```
QString const & Parameters::getFormatDate ( ) const
```

Renvoie

le format d'affichage de date

7.11.3.17 getGraphicsMaxPoints()

```
unsigned Parameters::getGraphicsMaxPoints ( ) const
```

Renvoie

le nombre maximal de points qu'affiche le graphique

7.11.3.18 openHelp()

```
bool Parameters::openHelp ( ) const
```

ouvre dans le navigateur par défaut la page d'aide de l'application

Renvoie

true si l'ouverture est un succès, false sinon

La documentation de cette classe a été générée à partir du fichier suivant :

- C:/Users/Jérémy/Desktop/projets/Budgettiseur/Budgettiseur/header/[Parameters.h](#)
- C:/Users/Jérémy/Desktop/projets/Budgettiseur/Budgettiseur/source/Parameters.cpp

7.12 Référence de la classe Ui::Principale

Est dérivée de [Ui_Principale](#).

Membres hérités additionnels

La documentation de cette classe a été générée à partir du fichier suivant :

- C:/Users/Jérémy/Desktop/projets/Budgettiseur/build-Budgettiseur-Desktop_Qt_5_15_0_MinGW_32_bit-Debug/ui_principale.h

7.13 Référence de la classe Principale

Est dérivée de QMainWindow.

Fonctions membres publiques

- **Principale** ([Parameters](#) &_parameters, QMainWindow *parent=nullptr)
- void **resizeEvent** (QResizeEvent *event) override

Connecteurs protégés

- void **resize** (int _index)
- void **apropos** ()
- void **save** ()
- void **help** ()
- void **activeVerification** ()
- void **addOperation** ()
- void **activeModification** ()
- void **activeVerificationMod** ()
- void **modOperation** ()
- void **delOperation** ()
- void **refreshTableau** (bool _init_recherches=false)
- void **refreshTableauBudget** (int _category)
- void **searchByNumero** (QString const &)
- void **searchByIntitule** (QString const &)
- void **activeVerificationBudget** ()
- void **activeOccurence** ()
- void **addBudget** ()
- void **chooseColor** ()
- void **getChosedColor** (QColor const &)
- void **modPrevision** (int _row)
- void **setPrevision** ()
- void **activeVerificationBudgetMod** ()
- void **activeOccurenceMod** ()
- void **modBudget** ()
- void **delBudget** ()
- void **chooseColorMod** ()
- void **getChosedColorMod** (QColor const &)
- void **changeDevise** (QString)
- void **changeDeviseVisible** (int _visible)
- void **changeFormatDate** ()
- void **changeMaxPoints** (int)
- void **changeWritePDF** (const QString &)
- void **changeWritePDF** ()
- void **selectWritePDF** ()
- void **changeWriteRegistry** (const QString &)
- void **changeWriteRegistry** ()
- void **selectWriteRegistry** ()
- void **changeWriteSave** (const QString &)
- void **changeWriteSave** ()
- void **selectWriteSave** ()

Fonctions membres protégées

- void **closeEvent** (QCloseEvent *event) override
- void **refreshListBudget** ()
- void **refreshTableauBudgetCategory** ()
- void **setUnsaved** ()

Attributs protégés

- `Ui::Principale * ui`
- `Calendar & calendar = Calendar::getInstance()`
- `Parameters & parameters`
- `QColor lastSelectionColor = QColor("black")`
- `Frequency * f = nullptr`

7.13.1 Documentation des constructeurs et destructeur

7.13.1.1 Principale()

```
Principale::Principale (
    Parameters & _parameters,
    QMainWindow * parent = nullptr )
```

Copyright

Lukos Budgeting Application copyright © 2020 Jérémy Godde

License :

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <https://www.gnu.org/licenses/>.

La documentation de cette classe a été générée à partir du fichier suivant :

- `C:/Users/Jérémy/Desktop/projets/Budgettiseur/Budgettiseur/principale.h`
- `C:/Users/Jérémy/Desktop/projets/Budgettiseur/Budgettiseur/principale.cpp`

7.14 Référence de la classe QCustomFrame

class `QCustomFrame` Représente un `QFrame` personnalisé permettant l'affichage d'un graphique

```
#include <customClass.h>
```

Est dérivée de `QFrame`.

Connecteurs publics

- void `focusOn` (int _point)
modifie l'indice de la donnée sur laquelle s'effectue le centrage du graphique

Fonctions membres publiques

- `QCustomFrame` (`QWidget *parent`)
constructeur
- `~QCustomFrame` ()=default
destructeur par défaut
- void `setBudgetCategory` (`BudgetCategory` const *const `_category`)
modifie le ponteur sur la catégorie contenant les données à afficher sous forme de graphique
- void `setMaxPoints` (unsigned const `_max_points`)
modifie le nombre de données à afficher sur le graphique

Fonctions membres protégées

- virtual void `paintEvent` (`QPaintEvent *`) override
événement personnalisé dessinant le graphique.

Attributs protégés

- `BudgetCategory` const * `category` = nullptr
pointeur sur la catégorie contenant les données à afficher sous forme de graphique
- int `margin_x` = 20
dimension des marges horizontales en px
- int `margin_y` = 20
dimension des marges verticales en px
- unsigned `point` = 0
indice de la donnée sur laquelle s'effectue le centrage du graphique
- unsigned `max_points` = 6
nombre de données à afficher sur le graphique

7.14.1 Description détaillée

class `QCustomFrame` Représente un `QFrame` personnalisé permettant l'affichage d'un graphique

Copyright

Lukos Budgeting Application copyright © 2020 Jérémie Godde

License :

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <https://www.gnu.org/licenses/>.

7.14.2 Documentation des constructeurs et destructeur

7.14.2.1 QCustomFrame()

```
QCustomFrame::QCustomFrame (
    QWidget * parent ) [explicit]
```

constructeur

Copyright

Lukos Budgeting Application copyright © 2020 Jérémy Godde

License :

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <https://www.gnu.org/licenses/>.

7.14.3 Documentation des fonctions membres

7.14.3.1 focusOn

```
void QCustomFrame::focusOn (
    int _point ) [slot]
```

modifie l'indice de la donnée sur laquelle s'effectue le centrage du graphique

Paramètres

<code>_point</code>	int : indice de la donnée
---------------------	---------------------------

7.14.3.2 setBudgetCategory()

```
void QCustomFrame::setBudgetCategory (
    BudgetCategory const *const _category )
```

modifie le ponteur sur la catégorie contenant les données à afficher sous forme de graphique

Paramètres

<code>_category</code>	<code>BudgetCategory</code> const* const : pointeur sur la catégorie
------------------------	--

7.14.3.3 setMaxPoints()

```
void QCustomFrame::setMaxPoints (
    unsigned const _max_points )
```

modifie le nombre de données à afficher sur le graphique

Paramètres

<code>_max_points</code>	unsigned const : nombre de données
--------------------------	------------------------------------

La documentation de cette classe a été générée à partir du fichier suivant :

- C:/Users/Jérémy/Desktop/projets/Budgettiseur/Budgettiseur/header/customClass.h
- C:/Users/Jérémy/Desktop/projets/Budgettiseur/Budgettiseur/source/customClass.cpp

7.15 Référence de la structure qt_meta_stringdata_Parameters_t

Attributs publics

- QByteArrayData **data** [15]
- char **stringdata0** [177]

La documentation de cette structure a été générée à partir du fichier suivant :

- C:/Users/Jérémy/Desktop/projets/Budgettiseur/build-Budgettiseur-Desktop_Qt_5_15_0_MinGW_64_bit-Debug/debug/moc_Parameters.cpp

7.16 Référence de la structure qt_meta_stringdata_Principale_t

Attributs publics

- QByteArrayData **data** [4]
- char **stringdata0** [44]

La documentation de cette structure a été générée à partir du fichier suivant :

- C:/Users/Jérémy/Desktop/projets/Budgettiseur/build-Budgettiseur-Desktop_Qt_5_15_0_MinGW_32_bit-Debug/debug/moc_principale.cpp

7.17 Référence de la structure qt_meta_stringdata_QCustomFrame_t

Attributs publics

- QByteArrayData **data** [4]
- char **stringdata0** [29]

La documentation de cette structure a été générée à partir du fichier suivant :

- C:/Users/Jérémy/Desktop/projets/Budgettiseur/build-Budgettiseur-Desktop_Qt_5_15_0_MinGW_64_bit-Debug/debug/moc_customClass.cpp

7.18 Référence de la classe Ui_Principale

Dérivée par [Ui::Principale](#).

Fonctions membres publiques

- void **setupUi** (QWidget *[Principale](#))
- void **retranslateUi** (QWidget *[Principale](#))
- void **setupUi** (QMainWindow *[Principale](#))
- void **retranslateUi** (QMainWindow *[Principale](#))

Attributs publics

- QGridLayout * **gridLayout**
- QHBoxLayout * **layoutPrincipale**
- QFrame * **layoutColonne**
- QVBoxLayout * **verticalLayout_3**
- QLabel * **labelIcône**
- QPushButton * **buttonPrincipale**
- QPushButton * **buttonBudget**
- QPushButton * **buttonCalendrier**
- QPushButton * **buttonParametres**
- QSpacerItem * **spacerColonne**
- QLabel * **labelDate**
- QVBoxLayout * **layoutBody**
- QLabel * **titre**
- QHBoxLayout * **layoutRecherche**
- QLabel * **labelNumeroRecherche**
- QComboBox * **comboNumeroRecherche**
- QLabel * **labelIntituleRecherche**
- QComboBox * **comboIntituleRecherche**
- QTableWidgetItem * **tableauOperations**
- QHBoxLayout * **layoutControle**
- QFormLayout * **LayoutControleGauche**
- QLabel * **labelIntitule**
- QLineEdit * **editIntitule**
- QLabel * **labelBudget**
- QComboBox * **comboBudget**
- QLabel * **labelDescription**
- QLineEdit * **editDescription**
- QProgressBar * **barProgression**
- QLabel * **labelProgression**
- QFormLayout * **layoutControleDroit**
- QDateEdit * **editDate**
- QDialogButtonBox * **boutonsValidation**
- QDoubleSpinBox * **editMontant**
- QLabel * **labelNumero**
- QLabel * **labelDateControle**
- QLabel * **labelMontant**
- QLineEdit * **editNumero**
- QAction * **actionSauvegarder**
- QAction * **action_propos**
- QAction * **actionExporter_en_PDF**
- QAction * **actionAide**
- QAction * **actionArchiver**
- QWidget * **centralWidget**
- QVBoxLayout * **verticalLayout**
- QTabWidget * **tabWidget**
- QWidget * **vuePrincipale**
- QGridLayout * **gridLayout_2**
- QLabel * **labelPeriode**
- QComboBox * **comboPeriode**
- QFrame * **line**
- QFrame * **line_3**
- QFrame * **line_2**
- QTabWidget * **tabOperation**
- QWidget * **ongletAjout**
- QSpacerItem * **verticalSpacer**
- QSpacerItem * **verticalSpacer_2**

```

— QWidget * ongletModif
— QHBoxLayout * horizontalLayout_7
— QFormLayout * LayoutControleGauche_2
— QLabel * labelIntituleMod
— QLineEdit * editIntituleMod
— QLabel * labelBudgetMod
— QComboBox * comboBudgetMod
— QLabel * labelDescriptionMod
— QLineEdit * editDescriptionMod
— QSpacerItem * verticalSpacer_3
— QHBoxLayout * horizontalLayout_6
— QSpacerItem * horizontalSpacer_3
— QPushButton * buttonSupprimerOperation
— QSpacerItem * horizontalSpacer_4
— QFormLayout * layoutControleDroit_2
— QDateEdit * editDateMod
— QDialogButtonBox * boutonsValidationMod
— QDoubleSpinBox * editMontantMod
— QLabel * labelNumeroMod
— QLabel * labelDateControleMod
— QLabel * labelMontantMod
— QLineEdit * editNumeroMod
— QSpacerItem * verticalSpacer_4
— QWidget * vueBudget
— QGridLayout * gridLayout_5
— QGridLayout * gridLayout_4
— QLabel * labelTitreBudget
— QLabel * labelTitreCategorie
— QTableWidgetItem * tableBudget
— QTableWidgetItem * tableBudgetCategory
— QFrame * line_5
— QPushButton * buttonBudgetPrevision
— QFormLayout * formLayout
— QLabel * labelBudgetPrevision
— QDoubleSpinBox * doubleBudgetPrevision
— QGroupBox * groupBudgetOccurrences
— QVBoxLayout * verticalLayout_2
— QRadioButton * radioBudgetJusteOccurence
— QRadioButton * radioBudgetFuturesOccurrences
— QLabel * labelBudgetOccurrences
— QCustomFrame * frameBudget
— QFrame * lineBudgetPrevision
— QTabWidget * tabBudget
— QWidget * ongletBudgetAjouter
— QFormLayout * controleBudget
— QLabel * label_3
— QLineEdit * editBudget
— QLabel * label_7
— QComboBox * comboBudgetType
— QLabel * label_4
— QDateEdit * editBudgetDate
— QLabel * label_5
— QComboBox * comboBudgetFrequence
— QLabel * label_6
— QComboBox * comboBudgetOccurence
— QPushButton * buttonBudgetCouleur
— QLabel * label_8
— QPushButton * buttonBudgetAjouter
— QWidget * ongletBudgetModif
— QFormLayout * formLayout_5
— QLineEdit * editBudgetMod
— QComboBox * comboBudgetTypeMod
— QPushButton * buttonBudgetCouleurMod
— QDateEdit * editBudgetDateMod
— QComboBox * comboBudgetFrequenceMod
— QComboBox * comboBudgetOccurenceMod
— QLabel * label_15
— QLabel * label_16
— QLabel * label_17
— QLabel * label_18
— QLabel * label_19
— QLabel * label_20
— QHBoxLayout * horizontalLayout
— QPushButton * buttonBudgetMod
— QPushButton * buttonBudgetSupprimer
— QLabel * labelTitre
— QFrame * line_4

```

```

— QWidget * tab
— QGridLayout * gridLayout_3
— QFrame * line_6
— QFormLayout * formLayout_4
— QLabel * label_11
— QLabel * label_12
— QLabel * label_13
— QLabel * label_14
— QHBoxLayout * horizontalLayout_3
— QLineEdit * lineParSave
— QPushButton * buttonParSave
— QHBoxLayout * horizontalLayout_4
— QLineEdit * lineParArchive
— QPushButton * buttonParArchive
— QHBoxLayout * horizontalLayout_5
— QLineEdit * lineParPDF
— QPushButton * buttonParPDF
— QFormLayout * formLayout_2
— QLabel * label_10
— QLabel * label
— QComboBox * comboParDevise
— QCheckBox * checkParDevise
— QGroupBox * groupBox
— QRadioButton * radioParFormatDate1
— QRadioButton * radioParFormatDate2
— QRadioButton * radioParFormatDate3
— QRadioButton * radioParFormatDate4
— QRadioButton * radioParFormatDate5
— QFormLayout * formLayout_3
— QLabel * label_2
— QSpinBox * spinParMaxPoints
— QLabel * label_9
— QFrame * line_7
— QMenuBar * menubar
— QMenu * menuFichier
— QStatusBar * statusbar

```

La documentation de cette classe a été générée à partir du fichier suivant :

```

— C:/Users/Jérémy/Desktop/projets/Budgettiseur/build-Budgettiseur-Desktop_Qt_5_15_0_MinGW_32_bit-↵
  Debug/ui_principale.h

```


Chapitre 8

Documentation des fichiers

8.1 Référence du fichier C:/Users/Jérémy/Desktop/projets/↵ Budgettiseur/Budgettiseur/header/Calendar.h

```
#include <QRgb>
#include <QtXml>
#include <QTextCodec>
#include "Container.h"
#include "Event.h"
```

Classes

- class [Operation](#)
class [Operation](#) Represente une opération
- class [Budget](#)
class [Budget](#) Represente un budget
- class [BudgetCategory](#)
class [BudgetCategory](#) Represente une catégorie budgétaire contenant plusieurs budgets
- class [Calendar](#)
class [Calendar](#) Represente un calendrier budgétaire contenant plusieurs catégories budgétaires

Fonctions

- `Q_DECLARE_METATYPE` ([Operation](#) *)
- `Q_DECLARE_METATYPE` ([BudgetCategory](#) *)
- `QString const operator<<` (`QString const &_s1`, `QString const &_s2`)
- `std::ostream & operator<<` (`std::ostream &_f`, `QString const &_s`)
- `std::ostream & operator<<` (`std::ostream &_f`, [Date](#) const &_d)
- `QString getAttributeFromXML` (`QXmlStreamReader *_xml`, `QString const _attr`)

8.1.1 Description détaillée

Copyright

Lukos Budgeting Application copyright © 2020 Jérémy Godde

License :

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <https://www.gnu.org/licenses/>.

Auteur

Jérémy Godde

Version

1

Date

july 2020

8.1.2 Documentation des fonctions

8.1.2.1 `getAttributeFromXML()`

```
QString getAttributeFromXML (
    QDomStreamReader * _xml,
    QString const _attr )
```

Paramètres

<code>_xml</code>	<code>QDomStreamReader*</code> : flux xml
<code>_attr</code>	<code>QString const&</code> : nom de l'attribut à retourner

Renvoie

la valeur de l'attribut rechercher dans le flux xml

8.1.2.2 `operator<<()` [1/3]

```
QString const & operator<< (
    QString const & _s1,
    QString const & _s2 )
```

Paramètres

<code>_s1</code>	<code>QString const&</code> : chaîne à laquelle concaténer
<code>_s2</code>	<code>QString const&</code> : chaîne à concaténer

Renvoie

une chaîne correspondant à la concaténation de deux chaînes

8.1.2.3 operator<<() [2/3]

```
std::ostream & operator<< (
    std::ostream & _f,
    Date const & _d )
```

Paramètres

↩ _↩ <i>f</i>	std::ostream& : sortie standard à laquelle concaténer
↩ _↩ <i>d</i>	Date const& : date à concaténer

Voir également

[Date](#)

Renvoie

une sortie standard correspondant à la concaténation d'une sortie standard et d'une date

8.1.2.4 operator<<() [3/3]

```
std::ostream & operator<< (
    std::ostream & _f,
    QString const & _s )
```

Paramètres

↩ _↩ <i>f</i>	std::ostream& : sortie standard à laquelle concaténer
↩ _↩ <i>s</i>	QString const& : chaîne à concaténer

Renvoie

une sortie standard correspondant à la concaténation d'une sortie standard et d'une chaîne

8.2 Référence du fichier C:/Users/Jérémy/Desktop/projets/↵ Budgettiseur/Budgettiseur/header/Container.h

```
#include "Exception.h"
```

Classes

- class `Container< T >`
class `Container< T >` Représente un contenant d'objets `T`
- class `Container< T >::Iterator`
class `Iterator` Représente l'itérateur de la classe `Container`

8.2.1 Description détaillée

Copyright

Lukos Budgeting Application copyright © 2020 Jérémy Godde

License :

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <https://www.gnu.org/licenses/>.

Auteur

Jérémy Godde

Version

1

Date

july 2020

8.3 Référence du fichier C:/Users/Jérémy/Desktop/projets/↵ Budgettiseur/Budgettiseur/header/Event.h

```
#include "Exception.h"  
#include <QVariant>  
#include <QDate>
```


Classes

- class [Frequency](#)
class [Frequency](#) Représente une fréquence de répétition.
- class [Date](#)
class [Date](#) Représente une date.
- class [Event](#)
class [Event](#) Représente un événement.

Fonctions

- **Q_DECLARE_METATYPE** ([Frequency](#) *)
- QString const [operator<<](#) (QString const &_s, [Frequency](#) const &_f)
- QString const [operator<<](#) (QString const &_s, [Date](#) const &_d)

8.3.1 Description détaillée

Copyright

Lukos Budgeting Application copyright © 2020 Jérémy Godde

License :

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <https://www.gnu.org/licenses/>.

Auteur

Jérémy Godde

Version

1

Date

july 2020

8.3.2 Documentation des fonctions

8.3.2.1 [operator<<\(\)](#) [1/2]

```
QString const operator<< (
    QString const &_s,
    Date const &_d )
```

Paramètres

↩ _↩ s	QString const& : chaîne à laquelle concaténer
↩ _↩ d	Date const& : date à concaténer

Voir également

[Date](#)

Renvoie

une chaîne correspondant à la concaténation d'une chaîne et d'une date

8.3.2.2 operator<<() [2/2]

```
QString const operator<< (
    QString const & _s,
    Frequency const & _f )
```

Paramètres

↩ _↩ s	QString const& : chaîne à laquelle concaténer
↩ _↩ f	Frequency const& : fréquence à concaténer

Voir également

[Frequency](#)

Renvoie

une chaîne correspondant à la concaténation d'une chaîne et d'une fréquence

8.4 Référence du fichier C:/Users/Jérémy/Desktop/projets/↩ Budgettiseur/Budgettiseur/header/Exception.h

```
#include <iostream>
#include <QString>
```

Classes

- class [Exception](#)
class [Exception](#) Represente une exception

8.4.1 Description détaillée

Copyright

Lukos Budgeting Application copyright © 2020 Jérémy Godde

License :

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <https://www.gnu.org/licenses/>.

Auteur

Jérémy Godde

Version

1

Date

july 2020

8.5 Référence du fichier C:/Users/Jérémy/Desktop/projets/Budgettiseur/Budgettiseur/header/Parameters.h

```
#include "Exception.h"
#include <QtXml>
#include <QTextCodec>
#include <QMessageBox>
#include <QDesktopServices>
```

Classes

- class [Parameters](#)
class [Parameters](#) Represente le gestionnaire des paramètres de l'application

8.5.1 Description détaillée

Copyright

Lukos Budgeting Application copyright © 2020 Jérémy Godde

License :

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <https://www.gnu.org/licenses/>.

Auteur

Jérémy Godde

Version

1

Date

july 2020

Index

- addBudgetCategory
 - Calendar, [30](#)
- addOperation
 - Budget, [20](#)
 - BudgetCategory, [25](#)
- begin
 - Container< T >, [34](#)
- Budget, [19](#)
 - addOperation, [20](#)
 - getBalance, [21](#)
 - getDiffBtwnBalances, [21](#)
 - getForecastBalance, [21](#)
 - load, [21](#)
 - operator<, [22](#)
 - save, [22](#)
 - setEdit, [23](#)
 - show, [23](#)
- BudgetCategory, [23](#)
 - addOperation, [25](#)
 - delOperation, [25](#)
 - getColor, [26](#)
 - getSumBalances, [26](#)
 - getType, [26](#)
 - load, [26](#)
 - save, [27](#)
 - setColor, [27](#)
 - setOccurence, [27](#)
 - setSubject, [28](#)
 - setType, [28](#)
 - show, [28](#)
- C:/Users/J  r  my/Desktop/projets/Budgettiseur/Budgettiseur/header/Calendar.h,
[69](#)
- C:/Users/J  r  my/Desktop/projets/Budgettiseur/Budgettiseur/header/Container.h,
[72](#)
- C:/Users/J  r  my/Desktop/projets/Budgettiseur/Budgettiseur/header/Event.h,
[72](#)
- C:/Users/J  r  my/Desktop/projets/Budgettiseur/Budgettiseur/header/Exception.h,
[74](#)
- C:/Users/J  r  my/Desktop/projets/Budgettiseur/Budgettiseur/header/Parameters.h,
[75](#)
- Calendar, [29](#)
 - addBudgetCategory, [30](#)
 - calendar, [33](#)
 - getBalance, [30](#)
 - getInstance, [31](#)
 - load, [31](#)
 - operator>>, [31](#)
 - remove, [32](#)
 - save, [32](#)
 - show, [32](#)
- calendar
 - Calendar, [33](#)
- Calendar.h
 - getAttributeFromXML, [70](#)
 - operator<<, [70](#), [71](#)
- changeDevise
 - Parameters, [55](#)
- changeDeviseVisible
 - Parameters, [55](#)
- changeFormatDate
 - Parameters, [56](#)
- changeMaxPoints
 - Parameters, [56](#)
- changeWritePDF
 - Parameters, [56](#)
- changeWriteRegistry
 - Parameters, [56](#)
- changeWriteSave
 - Parameters, [57](#)
- Container
 - Container< T >, [34](#)
- Container< T >, [33](#)
 - begin, [34](#)
 - Container, [34](#)
 - getSize, [35](#)
 - insert, [35](#)
 - operator[], [35](#), [36](#)
 - remove, [36](#)
- Container< T >::Iterator, [49](#)
 - isTerminated, [50](#)
 - operator*, [50](#)
 - operator++, [50](#)
- Date, [37](#)
- addEvent, [38](#)
- fromString, [38](#)
- getDay, [38](#)
- getMonth, [38](#)
- getYear, [38](#)
- getToday, [39](#)
- operator<, [40](#)
- operator+, [39](#)
- operator-, [39](#)
- toQDate, [40](#)
- toString, [41](#)

- delOperation
 - BudgetCategory, [25](#)
- Event, [41](#)

- getDate, [42](#)
- getFrequency, [42](#)
- getOccurence, [42](#)
- getSubject, [43](#)
- operator<, [43](#)
- operator>, [44](#)
- Event.h
 - operator<<, [73](#), [74](#)
- Exception, [45](#)
 - getStdString, [45](#)
 - getString, [45](#)
- focusOn
 - QCustomFrame, [63](#)
- Frequency, [46](#)
 - Frequency, [46](#)
 - fromString, [47](#)
 - getFrequency, [47](#)
 - getNbMonth, [47](#)
 - getNbWeek, [48](#)
 - getNbYear, [48](#)
 - getRythm, [48](#)
 - operator*, [48](#)
 - toString, [49](#)
- fromString
 - Date, [38](#)
 - Frequency, [47](#)
- getAppLongName
 - Parameters, [57](#)
- getAppName
 - Parameters, [57](#)
- getAppVersion
 - Parameters, [57](#)
- getAttributeFromXML
 - Calendar.h, [70](#)
- getBalance
 - Budget, [21](#)
 - Calendar, [30](#)
- getColor
 - BudgetCategory, [26](#)
- getDate
 - Event, [42](#)
 - Operation, [51](#)
- getDay
 - Date, [38](#)
- getDescription
 - Operation, [52](#)
- getDevise
 - Parameters, [58](#)
- getDeviseVisible
 - Parameters, [58](#)
- getDiffBtwnBalances
 - Budget, [21](#)
- getDirPDF
 - Parameters, [58](#)
- getDirRegistry
 - Parameters, [58](#)
- getFileSave
 - Parameters, [58](#)
- getForecastBalance
 - Budget, [21](#)
- getFormatDate
 - Parameters, [59](#)
- getFrequency
 - Event, [42](#)
 - Frequency, [47](#)
- getGraphicsMaxPoints
 - Parameters, [59](#)
- getInstance
 - Calendar, [31](#)
- getMonth
 - Date, [38](#)
- getNbMonth
 - Frequency, [47](#)
- getNbWeek
 - Frequency, [48](#)
- getNbYear
 - Frequency, [48](#)
- getOccurence
 - Event, [42](#)
- getReference
 - Operation, [52](#)
- getRythm
 - Frequency, [48](#)
- getSize
 - Container< T >, [35](#)
- getStdString
 - Exception, [45](#)
- getString
 - Exception, [45](#)
- getSubject
 - Event, [43](#)
 - Operation, [52](#)
- getSumBalances
 - BudgetCategory, [26](#)
- getToday
 - Date, [39](#)
- getType
 - BudgetCategory, [26](#)
- getValue
 - Operation, [52](#)
- getYear
 - Date, [39](#)
- insert
 - Container< T >, [35](#)
- isTerminated
 - Container< T >::Iterator, [50](#)
- load
 - Budget, [21](#)
 - BudgetCategory, [26](#)
 - Calendar, [31](#)
 - Operation, [52](#)
- openHelp
 - Parameters, [59](#)

- Operation, 50
 - getDate, 51
 - getDescription, 52
 - getReference, 52
 - getSubject, 52
 - getValue, 52
 - load, 52
 - operator<, 53
 - save, 53
- operator<
 - Budget, 22
 - Date, 40
 - Event, 43
 - Operation, 53
- operator<<
 - Calendar.h, 70, 71
 - Event.h, 73, 74
- operator>
 - Event, 44
- operator>>
 - Calendar, 31
- operator*
 - Container< T >::Iterator, 50
 - Frequency, 48
- operator+
 - Date, 39
- operator++
 - Container< T >::Iterator, 50
- operator-
 - Date, 39
- operator[]
 - Container< T >, 35, 36
- Parameters, 53
 - changeDevise, 55
 - changeDeviseVisible, 55
 - changeFormatDate, 56
 - changeMaxPoints, 56
 - changeWritePDF, 56
 - changeWriteRegistry, 56
 - changeWriteSave, 57
 - getAppLongName, 57
 - getAppName, 57
 - getAppVersion, 57
 - getDevise, 58
 - getDeviseVisible, 58
 - getDirPDF, 58
 - getDirRegistry, 58
 - getFileSave, 58
 - getFormatDate, 59
 - getGraphicsMaxPoints, 59
 - openHelp, 59
 - Parameters, 55
- Principale, 60
 - Principale, 61
- QCustomFrame, 61
 - focusOn, 63
 - QCustomFrame, 62
 - setBudgetCategory, 63
 - setMaxPoints, 63
 - qt_meta_stringdata_Parameters_t, 64
 - qt_meta_stringdata_Principale_t, 64
 - qt_meta_stringdata_QCustomFrame_t, 64
- remove
 - Calendar, 32
 - Container< T >, 36
- save
 - Budget, 22
 - BudgetCategory, 27
 - Calendar, 32
 - Operation, 53
- setBudgetCategory
 - QCustomFrame, 63
- setColor
 - BudgetCategory, 27
- setEdit
 - Budget, 23
- setMaxPoints
 - QCustomFrame, 63
- setOccurence
 - BudgetCategory, 27
- setSubject
 - BudgetCategory, 28
- setType
 - BudgetCategory, 28
- show
 - Budget, 23
 - BudgetCategory, 28
 - Calendar, 32
- toQDate
 - Date, 40
- toString
 - Date, 41
 - Frequency, 49
- Ui, 17
 - Ui::Principale, 59
 - Ui_Principale, 65