

□ □ (https://profile.intra.42.fr/searchs)

проследить

(https://profile.intra.42.fr)

ШКАЛА ДЛЯ ПРОЕКТА CPP MODULE 02 (/ПРОЕКТЫ/CPP-МОДУЛЬ-02)

Вы должны оценить 1 студента в этой команде

□

Git-репозиторий

git@vogsphere.msk.21-school.ru:vogsphere/intra-uuid-c9bd8ce □

Введение

Пожалуйста, соблюдайте следующие правила:

- Оставайтесь вежливым, учтивым, уважительным и конструктивным на протяжении всего процесса оценки. От этого зависит благополучие общества.
- Определите вместе со студентом или группой, чья работа оценивается, возможные недостатки в их проекте. Найдите время, чтобы обсудить и обсудить проблемы, которые, возможно, были выявлены.
- Вы должны учитывать, что могут быть некоторые различия в том, как ваши коллеги могли понять инструкции проекта и объем его функций. Всегда будьте непредвзяты и оценивайте их как можно честнее. Педагогика полезна только и только в том случае, если экспертная оценка проводится серьезно.

Методические рекомендации

- Оценивайте только ту работу, которая была сдана в репозиторий Git оцениваемого учащегося или группы.
- Дважды проверьте, что репозиторий Git принадлежит учащимся. Убедитесь, что проект соответствует ожиданиям. Также убедитесь, что «git clone» используется в пустой папке.
- Внимательно проверьте, чтобы не использовались вредоносные псевдонимы, чтобы обмануть вас и заставить оценить что-то, что не является содержимым официального репозитория.
- Чтобы избежать каких-либо неожиданностей и, если применимо, просмотрите вместе все используемые сценарии.

для облегчения выставления оценок (скрипты для тестирования или автоматизации).

- Если вы не выполнили задание, которое собираетесь оценивать, вы должны прочитать всю тему до начала процесса оценивания.

- Используйте доступные флаги, чтобы сообщить о пустом репозитории, неработающей программе, ошибке Norm, читерстве и так далее.

В этих случаях процесс оценки завершается и окончательная оценка равна 0 или -42 в случае списывания. Однако, за исключением списывания, учащимся настоятельно рекомендуется вместе просмотреть сданную работу, чтобы выявить ошибки, которые не следует повторять в будущем.

- Вам никогда не придется редактировать какой-либо файл, кроме файла конфигурации, если он существует. Если вы хотите отредактировать файл, найдите время, чтобы объяснить причины оцениваемому учащемуся и убедитесь, что вы оба согласны с этим.

- Также необходимо проверить отсутствие утечек памяти. Любая память, выделенная в куче, должна быть должным образом освобождена до окончания выполнения.

Вам разрешено использовать любой из различных инструментов, доступных на компьютере, таких как утечки, valgrind или e_fence. В случае утечек памяти отметьте соответствующий флаг.

Вложения

□ subject.pdf (<https://cdn.intra.42.fr/pdf/pdf/41373/en.subject.pdf>)

Предварительные испытания

Если подозревается мошенничество, оценка здесь останавливается. Используйте флаг «Обман», чтобы сообщить об этом. Принимайте это решение спокойно, взвешенно и, пожалуйста, используйте эту кнопку с осторожностью.

Предпосылки

Код должен компилироваться с помощью C++ и флагов -Wall -Wextra -Werror Не забывайте, что этот проект должен соответствовать стандарту C++98. Таким образом, функции или контейнеры C++ 11 (и более поздних версий) НЕ ожидаются.

Любое из этих значений означает, что вы не должны оценивать рассматриваемое

упражнение: - Функция реализована в заголовочном файле (кроме шаблонных функций).

- Makefile компилируется без необходимых флагов и/или другого компилятора, кроме c++.

Любое из этих действий означает, что вы должны пометить проект как «Запрещено».

Функция":

- Использование функции "C" (*alloc, *printf, бесплатно).

- Использование функции, не разрешенной в инструкции по упражнению.

- Использование «используя пространство имен» или ключевое слово "друг".
- Использование внешней библиотеки или функций из версий, отличных от C++98.

☐ Да☐ Нет

Ex00: Мой первый класс в православной канонической форме

В этом упражнении вводится понятие канонического класса на простом арифметическом примере: числа с фиксированной точкой.

Makefile

Существует Makefile, который компилируется с использованием соответствующих флагов.

☐ Да☐ Нет

Аксессуар

Класс Fixed (или его имя) должен предоставлять методы доступа к исходное значение:

- int getRawBits(void) const;

- недействительным setRawBits(

целое константное сырое);

Присутствуют ли эти функции-члены и функционируют ли они?

☐ Да☐ Нет

Канонический

Канонический класс должен предоставлять как минимум:

- Конструктор по умолчанию

- Разрушитель

- Конструктор копирования

- оператор присваивания копирования

Имеются ли эти элементы и функционируют ли они?

☐ Да☐ Нет

Ex01: На пути к более полезному числу с фиксированной точкой сорт

Ex00 был хорошим началом, но наш класс по-прежнему довольно бесполезен, поскольку он может представлять только значение с фиксированной точкой 0,0.

Makefile

Существует Makefile, который компилируется с использованием соответствующих флагов.

☐ Да

☐ Нет

Конструктор с плавающей запятой

Можно ли создать экземпляр из значения с плавающей запятой?

☐ Да

☐ Нет

<< оператор

Существует ли перегрузка оператора << и работает ли она?

☐ Да

☐ Нет

Значение фиксированной точки в целочисленное значение

Функция-член "int toInt(void) const;" который преобразует

значение с фиксированной запятой в целочисленное значение должно присутствовать. Это функционально?

☐ Да

☐ Нет

Значение с фиксированной запятой в значение с плавающей запятой

Функция-член "float toFloat(void) const;", которая преобразует

должно присутствовать значение с фиксированной запятой для значения с плавающей запятой. Это функционально?

☐ Да

☐ Нет

Целочисленный конструктор

Можно ли построить экземпляр из целочисленного значения?

☐ Да

☐ Нет

Ex02: Теперь мы говорим

В этом упражнении к классу добавляются функции сравнения и арифметики.

Makefile

Существует Makefile, который компилируется с использованием соответствующих флагов.

☐ Да

☐ Нет

Операторы сравнения

Правильно ли реализованы и работают шесть операторов сравнения (>, <, >=, <=, == и !=)?

☐ Да

☐ Нет

Арифметические операторы

Правильно ли реализованы и работают шесть арифметических операторов (+, -, *, * и /)?

☐ Да

☐ Нет

Другие операторы

Операторы преинкремента, постинкремента, предкремента и постдекремента реализованы и работают правильно?

☐ Да

☐ Нет

Статические перегрузки функций-членов

И последнее, но не менее важное: протестируйте статические функции-члены min() и max().
реализованы и работают исправно.

☐ Да

☐ Нет

Ex03: БСП

Это упражнение должно помочь вам осознать, насколько легко реализовать сложные алгоритмы, когда основы работают как надо.
предназначена.

Makefile

Существует Makefile, который компилируется с использованием соответствующих флагов.

☐ Да

☐ Нет

Балл класса

Существует класс Point, который имеет два атрибута (x и y) типа Fixed const.

У него также есть конструктор, который принимает два числа с плавающей запятой и инициализирует x и y с помощью
эти значения.

☐ Да

☐ Нет

Функция bsp

Есть функция bsp(), прототип которой

"bool bsp(Point const a, Point const b, Point const c, Point const point)".

Функция возвращает True, если точка находится внутри треугольника, описанного вершинами a, b и c.

В противном случае он возвращает False.

☐ Да

☐ Нет

Основное и тесты

По крайней мере, есть main для проверки того, что функция bsp() работает должным образом.

Запустите несколько тестов, чтобы убедиться в правильности возвращаемого значения.

☐ Да

☐ Нет

Рейтинги

Не забудьте проверить флаг, соответствующий защите

☐ Хорошо

☐ Пустая работа

☐ Незавершенная работа

☐ Недопустимая компиляция

☐ Чит

☐ Сбой

☐ Запрещенная функция

Заключение

Оставьте комментарий к этой оценке

Завершить оценку

Политика конфиденциальности (<https://signin.intra.42.fr/legal/terms/5>)

Условия использования видеонаблюдения (<https://signin.intra.42.fr/legal/terms/1>)

Правила процедуры (<https://signin.intra.42.fr/legal/terms/4>)

Декларация об использовании файлов cookie (<https://signin.intra.42.fr/legal/terms/2>)

Общие условия использования сайта (<https://signin.intra.42.fr/legal/terms/6>)

Правовые уведомления (<https://signin.intra.42.fr/legal/terms/3>)