

Дисциплина «Основы программирования на C++»

Практическое задание №1

29 апреля – 15 мая

«Работа с графами»

Процесс выполнения этого задания состоит из трёх частей:

- 1) реализация программы, согласно описанным в условии требованиям;
- 2) оценивание работ других студентов;
- 3) период «споров».

Период реализации программы:

После выдачи задания Вам необходимо выполнить его и загрузить архив (*.zip) с решением задачи (полностью заархивировать решение, созданное средой разработки) до крайнего срока. В работе строго запрещается указывать ФИО, а также любую другую информацию, которая может выдать авторство работы. В случае выявления факта деанонимизации работы, работа может быть аннулирована.

Период взаимного оценивания:

После окончания срока, отведенного на реализацию программы, начинается период взаимного оценивания. Вам будет необходимо проверить четыре-пять работ других студентов, также выполнявших данное задание, согласно критериям оценивания. Проверка осуществляется анонимно: Вы не знаете, чью работу Вы проверяете, также, как и человек, кому принадлежит решение, не знает, кем была проверена его работа. Помимо оценки Вам необходимо указать комментарий к каждому из критериев. В случае, если Вы снижаете балл, необходимо подробно описать, за что именно была снижена оценка. Также рекомендовано писать субъективные комментарии, связанные с тонкостями программной реализации, предлагать автору работу более оптимальные на ваш взгляд решения. Снимать баллы за субъективные особенности реализации запрещено.

Период споров:

По окончании периода взаимного оценивания, в случае если Вы не согласны с оценкой, выставленной одним из проверяющих, Вы можете вступить с этим студентом в анонимный диалог с целью уточнения причин выставления оценки по тому или иному критерию. В случае, если проверяющий не ответил Вам, или вы не пришли к обоюдному решению об изменении оценки, Вы можете поставить флаг, и работа будет рассмотрена одним из преподавателей.

Флаги, поставленные без предварительного обсуждения с проверяющим или после окончания периода выставления флагов, будут отклонены.

Также допустима перепроверка Вашей работы преподавателем. В таком случае оценки других проверявших работу не учитываются.

Дедлайн загрузки работы: 15 мая 23:59:00

Дедлайн проверки: 18 мая 23:59:00

Дедлайн обсуждения оценок: 20 мая 23:59:00

Дедлайн выставления флагов: 21 мая 23:59:00

Возможность поздней сдачи работы в этом задании предоставляться не будет.

Оценивание:

$O_{\text{итог}} = 0,8 * O_{\text{задание}} + 0,2 * O_{\text{проверки}}$, где $O_{\text{задание}}$ – неокруглённая десятибалльная оценка за решение задания выставленная проверяющими с учётом возможной перепроверки преподавателем, а $O_{\text{проверки}}$ неокруглённая десятибалльная оценка, выставленная студентами, чьи работы вы проверяли. Также в случае, если преподаватель обнаружит, что Вы проверили работы некачественно к Вам могут быть применены санкции в виде штрафа до 3 десятичных баллов от $O_{\text{итог}}$ (в таком случае оценка вычисляется по формуле $O_{\text{итог}} = O_{\text{задание}} - \text{Штраф}$).

Необходимо разработать консольное приложение – «Работа с графами».

Алгоритм работы приложения:

Приложение обеспечивает возможность работы с ориентированными и неориентированными невзвешенными графами без кратных ребер и петель.

Описанный ниже функционал необходимо реализовать в виде консольного меню.

В приложении должны поддерживаться следующие способы хранения графа:

1. Матрица смежности
2. Матрица инцидентности
3. Список смежности
4. Список ребер

В приложении должны поддерживаться следующие способы ввода и вывода графа и результатов (формат определяется самостоятельно, но должен быть понятен пользователю и описан):

1. Консольный ввод/вывод

2. Ввод/вывод в текстовый файл

Должны быть доступны следующие функции для работы с графами:

1. Подсчет степеней вершин для неориентированного графа и полустепеней вершин для ориентированного графа.
2. Подсчет количества ребер (для неориентированного)/дуг (для ориентированного) графа.
3. Вывод графа в другом представлении (например, если граф был сохранен в программе в виде матрицы смежности, то должна быть возможность его вывести и сохранить не только в виде матрицы смежности, но и в виде матрицы инцидентности, списка смежности, списка ребер).

В качестве дополнительного функционала должно быть реализовано (для получения оценки больше 8):

1. хранение графа в представлениях FO и FI, MFO и MFI, BFO и BMFO. Весь функционал из условия должен работать на этих представлениях;
2. функции обхода графа в глубину и ширину (DFS и BFS). Обход в глубину должен быть реализован в двух видах – рекурсивно и не рекурсивно.

Дополнительные требования (критерии оценивания):

1. Для проверки в систему PeerGrade должен быть загружен архив, содержащий в себе файл CMakeList.txt, с решением, выполненным в среде разработки CLion (версия языка C++ – 20). Ожидается, что проверка работы будет проводиться, в среде разработки CLion, поэтому в случае выполнения задания с использованием другой среды разработки настоятельно рекомендуется проверить возможность открытия и запуска проекта в CLion.
2. Текст программы должен быть отформатирован согласно кодстайлу курса¹.
3. Программа не должна завершаться аварийно (или уходить в бесконечный цикл) при корректных входных данных. Формат входных данных должен быть строго определён разработчиком и выводиться в качестве подсказок в программе или быть описан в readme файле.
4. Программа должна быть декомпозирована. Каждый из логических блоков должен быть выделен в отдельный метод. Не строго, но желательно, чтобы каждый метод по длине не превышал 40 строк.
5. Интерфейс программы должен быть понятен. Пользователю должны выводиться подсказки о возможных дальнейших действиях и иные

¹ https://edu.hse.ru/pluginfile.php/1597597/mod_resource/content/1/style.pdf

необходимые сообщения. Предполагается, что для успешного использования программы не требуется обращения к исходному коду программы.

6. Должно быть реализовано повторение решения. То есть, после окончания выполнения очередной функции, пользователю предлагается завершить программу или начать игру снова.
7. Текст программы должен быть документирован. Названия переменных и методов должны быть на английском языке и отражать суть хранимых значений / выполняемых действий.

8. Работа должна быть выполнена в соответствии с заданием, критерии оценивания (“Rubric”) доступны для пред-просмотра в системе “PeerGrade”.
9. Для получения отличной оценки более 8 студенту необходимо реализовать дополнительный функционал описанный в задании выше.
10. Также оценивается общее впечатление, которое производит работа (как с точки зрения пользовательского интерфейса, так и с точки зрения написания кода программы). Эта часть оценки остаётся на усмотрение проверяющего.