Время выполнения задания

3 часа первая часть + 3 часа вторая часть(задания будут выданы позднее)

Задание

Разработать приложение для игры в Кингсбург.

Условия выполнения работы

Допускается покидать рабочее место не более чем на 10 минут 1 раз в час.

В случае использования недопустимой (см. ниже) аппаратуры и информационных ресурсов, а также замеченного общения с другими людьми преподаватель вправе не засчитывать отдельные либо все части выполненного задания, а также учащийся может быть временно удалён с экзамена с проставлением оценки “неудовлетворительно” в ведомость.

Разрешается использовать	***Не допускается***

1. Программное обеспечение: ● Любую среду разработки на C++ 14 и позднее. ● Текстовый редактор. 2. Справочные ресурсы по языку C++: ● https://msdn.microsoft.com, ● https://cppreference.com 3. Печатные книги и учебники по языку C++. 4. Личные конспекты студента.	Использовать ● Прочие программы и веб-страницы, ● Съёмные носители информации, ● Материалы на сервере Московского Политеха. Общаться с другими учащимися очно или с использованием дистанционных технологий.

Порядок выполнения работы

1) Работа выполняется на основной ОС рабочего ПК либо в виртуальной машине.
2) На рабочем месте установлено все необходимое программное обеспечение:

- VisualStudio версии не ниже 2017, в комплектации, включающей средства разработки консольных приложений на C++.
- QT версии не ниже 5.12, в комплектации, включающей средства разработки консольных приложений на C++.
- Или браузер.

Критерии оценки (техническая глава)

Проект приложения в среде разработки должен состоять из номера группы и фамилии учащегося латиницей, по образцу

“211_352_Ivanov”. По окончании работы проект отправляется на почту преподавателей и в удаленный репозиторий.

Любой ввод данных в приложении осуществляется программно, ручной ввод с консоли не используется.

По любому фрагменту исходного кода разработанного приложения могут быть заданы вопросы. К примеру

- описать назначение выбранной строки кода или объявленной переменной,
- продемонстрировать настройки, сделанные для подключения библиотек и т.д.

Ответ должен быть получен без подготовки. В случае, если учащийся не ответил на вопрос за 2 минуты, преподаватель вправе не оценивать фрагмент задания, к которому относился вопрос.

Текст программы должен быть оформлен в однообразном и легко читаемом стиле:

- Идентификаторы переменных должны иметь осмысленное название.
- Каждый базовый блок кода имеет одинаковый отступ.
- Строки кода не длиннее 100 символов.

Критерии оценки

За лабораторные и тесты в семестре можно получить до 50 баллов. За задание практического экзамена - до 40 баллов. За итоговое теоретическое тестирование - до 10 баллов.

Баллы	***Оценка***
65-74	3
75-84	4
>=85	5

Данная таблица носит рекомендательный характер. Оценка может быть изменена в сторону повышения и в сторону снижения по усмотрению преподавателя.

Задание Бакалаврам

Задание	**API**	**Баллы**
Первая часть экзамена		

Написан класс для описания текущего состояния игры Кингсбург.
Корректно прописаны конструктор по умолчанию, деструктор, инициализирующий конструктор (параметр - количество игроков) и копирующий конструктор, а так же все методы из задания экзамена с заглушкой вместо реализации.
 	
Своиства класса:
1) список игроков с учетом очередности хода (имя игрока,имеющиеся ресурсы (золото, дерево, камень, жетоны, кубик1, кубик2, кубик3, доп.кубик), наличие советника, значение воинского реестра, количество победных очков, вектор построек);
2) вектор с советниками (имя советника; имя игрока, который занял);
3) год (от 1 до 5)
	class Game_Surname
	{
	…
	}
		5

4) фаза (от 1 до 8) 5) враг в текущем году Можно использовать вспомогательные структуры (самописные или библиотечные). В названии класса и файлов указать вашу фамилию. Результат - файлы game_surname.h, game_surname.cpp		
В репозиторий прикреплены скриншоты вашей работы с git с помощью консоли. Скриншоты можно загружать через web-интерфейс. Результат - файлы *.jpg		2
Работа отправлена в свой удаленный репозиторий. Репозиторий содержит только файлы, необходимые для постройки проекта на любой сторонней машине и не содержит временные и автоматически перестраиваемые файлы.		2
Написано консольное приложение для демонстрации работы класса, которое		3

реализует многопользовательскую игру. Source_Surname.cpp		
Реализован метод класса или функция (при отсутствии класса), описывающий фазу 1.	void Game_Surname::phase1 (); или void phase1(SomeDataStruct& curr_game);	2
Реализован метод класса или функция (при отсутствии класса), описывающий фазу 3.	void Game_Surname::phase3 (); или void phase3(SomeDataStruct& curr_game);	1
Реализован метод класса или функция (при отсутствии класса), описывающий фазу 5.	void Game_Surname::phase5 (); или void phase5(SomeDataStruct& curr_game);	1
Реализован метод класса или функция (при отсутствии класса), описывающий фазу 7.	void Game_Surname::phase7 (); или	1

	void phase7(SomeDataStruct& curr_game);	
Реализован метод класса или функция (при отсутствии класса), описывающий фазу 8. Для применения результатов боя к игроку использовать функцию-заглушку.	void Game_Surname::phase8 (); или void phase8(SomeDataStruct& curr_game);	2
Реализована функция, применяющая результат боя к параметрам игрока по плану: 1) рассчитывающая общее значение обороны игрока с учетом помощи короля, построек и вида врага. 2) применение последствий боя к игроку.	void defense_level (const char* enemy_name, int king_help, SomeDataStructForPlayer& player);	2
Вторая часть экзамена		
Реализован метод класса или функция (при отсутствии класса), описывающий фазы 2, 4,6, вместо этапов бросков кубика, изменения порядка ходов, влияния на советников, помощи советников, возведения зданий и применения эффектов зданий	void Game_Surname::phase246(); или void phase246(SomeDataStruct& curr_game);	1

должны быть функции-заглушки или методы-заглушки.		

Реализован метод класса или функция (при отсутствии класса), описывающий броски кубиков игроками и изменение порядка ходов игроков в фазах 2, 4,6.	void Game_Surname::phase246_bones();
	или
	void phase246_bones(SomeDataStruct& curr_game);
		2

Реализован метод класса или функция (при отсутствии класса), описывающий влияния игроков на королевских советников и помощь советников в фазах 2, 4,6. Помощь советника описывается функцией-заглушкой.	void Game_Surname::phase246_advisor();
	или
	void phase246_advisor(SomeDataStruct& curr_game);
		2

Реализована функция, описывающая помощь советника.	void advisor_help(const char* advisor_name, SomeDataStructForPlayer& player);	4
Реализована функция, описывающая возведение	void building(const char* advisor_name,	4

зданий игроком и применение эффектов зданий к игроку.	SomeDataStructForPlayer& player);	

Реализован метод класса или функция (при отсутствии класса) для загрузки игры из файла.	bool Game_Surname::load_game(const char* filename);
	или
	SomeDataStruct load_game(Const char* filename);
	или
	bool load_game(Const char* filename, SomeDataStruct& curr_game);
		3

Реализован метод класса или функция (при отсутствии класса) для сохранения игры в файл.	bool Game_Surname:: save_game(const char* filename);
	или
	bool save_game(const char* filename, const SomeDataStruct& curr_game);
		3

Правила игры
https://www.mosigra.ru/image/data/mosigra.product.other/401/072/kingsburg.pdf