



**Оглавление**

[ЦЕЛЬ ВЫПОЛНЕНИЯ 3](#_heading=h.gjdgxs)

[НАЗНАЧЕНИЕ РАЗРАБОТКИ 3](#_heading=h.30j0zll)

[ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ 3](#_heading=h.1fob9te)

[СТАДИИ И ЭТАПЫ РАЗРАБОТКИ 8](#_heading=h.3znysh7)

[ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ 9](#_heading=h.2et92p0)

# ЦЕЛЬ ВЫПОЛНЕНИЯ

Целью работы является создание веб-приложения для игры в классические шахматы. Веб-приложение должно быть размещено на сервере и иметь открытый доступ для пользователей. Приложение состоит из frontend части, без взаимодействия с сервером. Все вычисления выполняются на стороне клиента.

В ходе работы предполагается изучение материалов по работе с React библиотекой для создания пользовательских интерфейсов.

Разработка ведется с языком программирования Typescript, который позволяет создавать крупные приложения со статической типизацией, что дает возможность развивать и добавлять новый функционал в приложение в дальнейшем. Typescript — язык программирования, представленный Microsoft в 2012 году и позиционируемый как средство разработки веб-приложений, расширяющее возможности JavaScript. Как называют его разработчики, typescript – это javascript, который масштабируется.

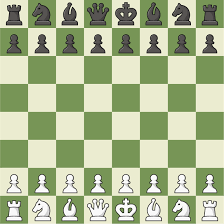
# НАЗНАЧЕНИЕ РАЗРАБОТКИ

Приложение будет находится в открытом доступе, предоставляя возможность посетителям сайта поиграть в шахматы.

# ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ

Приложение должно содержать в себе следующие элементы:

* Игровая доска – Поле для игры в шахматы 8 на 8 клеток, на котором расположены игровые фигуры(рис.1).

  
рис.1 Шахматная доска

* Фигуры – должны перемещаться согласно правилам игры в шахматы. Пользователь видит все возможные варианты ходов при выборе фигуры.  
    
  Список фигур:
  + Король (King)
  + Королева (Queen)
  + Слон (Bishop)
  + Конь (Knight)
  + Ладья (Rook)
  + Пешка (Pawn)
* Игровые часы – указывают время каждого игрока, имеется кнопка для перезапуска игры.
* Поле с информацией о партии – содержит информацию о забранных фигурах для каждого игрока.

Приложение должно работать на стороне клиента. Для работы не должны отправляться запросы на сервер. Предполагается возможность локального запуска без доступа к сети интернет.

Скорость развертывания приложения должна составлять не более 3000мс.

Первый ход всегда осуществляется белой стороной.

Ограничение по времени составляет 5 минут для каждой стороны, без добавления.

* Переносимость: способность программного обеспечения работать с несколькими аппаратными платформами или операционными системами. Обеспечивается благодаря использованию высокоуровневых языков программирования, сред разработки и выполнения, поддерживающих условную компиляцию, компоновку и выполнение кода для различных платформ.
* Надёжность: свойство объекта сохранять во времени в установленных пределах значения всех параметров, характеризующих способность выполнять требуемые функции в заданных условиях применения, технического обслуживания, хранения и транспортирования.  
  Интуитивно надёжность объектов связывают с недопустимостью отказов в работе. Это есть понимание надёжности в «узком» смысле — свойство объекта сохранять работоспособное состояние в течение некоторого времени или некоторой наработки. Иначе говоря, надёжность объекта заключается в отсутствии непредвиденных недопустимых изменений его качества на стадии эксплуатации (при его использовании, обслуживании, хранении, транспортировании). Надёжность — комплексное свойство, которое в зависимости от назначения объекта и условий его эксплуатации может включать в себя свойства безотказности, долговечности, ремонтопригодности и сохраняемости, а также определённое сочетание этих свойств.  
  Для количественной оценки надёжности используют так называемые единичные показатели надёжности (характеризуют только одно свойство надёжности) и комплексные показатели надёжности (характеризуют несколько свойств надёжности в определённом интервале времени).
* Эффективность: способность выполнять работу и достигать необходимого или желаемого результата с наименьшей затратой времени и усилий.
* Удобство использования: (от англ. usability — «удобство и простота использования, степень удобства использования»), также удобство использования, пригодность использования, эргономичность — способность продукта быть понимаемым, изучаемым, используемым и привлекательным для пользователя в заданных условиях (ISO/IEC 25010); свойство системы, продукта или услуги, при наличии которого конкретный пользователь может эксплуатировать систему в определённых условиях для достижения установленных целей с необходимой результативностью, эффективностью и удовлетворённостью (ISO 9241-210). Удобство (пригодность) использования системы не сводится только к тому, насколько её легко эксплуатировать. В соответствии со стандартами серии ISO 9241 эту характеристику следует понимать более широко, учитывая личные цели пользователя, его эмоции и ощущения, связанные с восприятием системы, а также удовлетворённость работой. Свойства, необходимые для обеспечения пригодности использования, зависят также от задачи и окружающей среды. Пригодность использования — не абсолютное понятие, оно может различным образом проявляться в определённых условиях эксплуатации.
* Сопровождаемость: процесс улучшения, оптимизации и устранения дефектов программного обеспечения (ПО) после передачи в эксплуатацию. Сопровождение ПО — это одна из фаз жизненного цикла программного обеспечения, следующая за фазой передачи ПО в эксплуатацию. В ходе сопровождения в программу вносятся изменения с тем, чтобы исправить обнаруженные в процессе использования дефекты и недоработки, а также для добавления новой функциональности, с целью повысить удобство использования (юзабилити) и применимость ПО. Сопровождение программного обеспечения стандартизовано, имеются национальные стандарты Российской Федерации, идентичные международным (ISO/IEC 12207:2008 System and software engineering — Software life cycle processes, ГОСТ Р ИСО/МЭК 12207-2010 «Национальный стандарт Российской Федерации. Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств»; ISO/IEC 14764:99 Information technology — Software maintenance, ГОСТ Р ИСО/МЭК 14764-2002 «Государственный стандарт Российской Федерации. Информационная технология. Сопровождение программных средств»; IEEE 1219).
* Тестируемость
* Понятность
* Модифицируемость
* Функциональная пригодность
* Уровень производительности

# СТАДИИ И ЭТАПЫ РАЗРАБОТКИ

Первым этапом утверждается внешний вид сайта:

* Фигуры
* Доска
* Иконка доступного хода
* Иконка доступного взятия фигуры

После утверждения внешнего вида, разрабатывается игровое поле. Это клеточная доска для игры в шахматы размером 8 на 8 клеток. Предполагается, что поле можно будет в дальнейшем использовать и для других игр (например, для игры в шашки).

Когда игровая доска готова, необходимо разместить фигуры в начальное положение. Также для фигур действует концепт двойного связывания, когда каждая клетка знает о фигуре, которая на ней находится, но и каждая фигура располагает информацией о своей клетке.

После того, как были размещены все фигуры, реализуется логика их перемещения. Общая логика перемещения описывается в классе Figure, а конкретные условия для отдельных фигур (King, Queen…) будут расширять класс Figure собственным классом с логикой.

# ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ

Итоговая работа должна соответствовать всем требованиям и покрыта тестами.

* Фигуры корректно перемещаются
* Часы функционируют
* Игру можно перезапустить
* Корректно отображаются потерянные фигуры

