Софийски университет „Св. Климент Охридски“

Факултет по математика и информатика

Курс по Обектно-ориентирано програмиране

на специалност Информатика

Летен семестър на учебната 2018/2019 година

Шахмат

Автор: Емил Христов Иваничков

ФН:45557

* 1. **Увод:**
  2. Описание и идея на проекта:

Приложението представлява конзолна игра на Шахмат. Целта на приложението е чрез него да бъде обучена невронна мрежа да играе играта. Освен това играта може да бъде използвана за създаване на турнири между студенти. Проекта трябва напълно и автентично да пресъздава играта и да извежда информация за класирането

* 1. Цел и задачи за разработка:

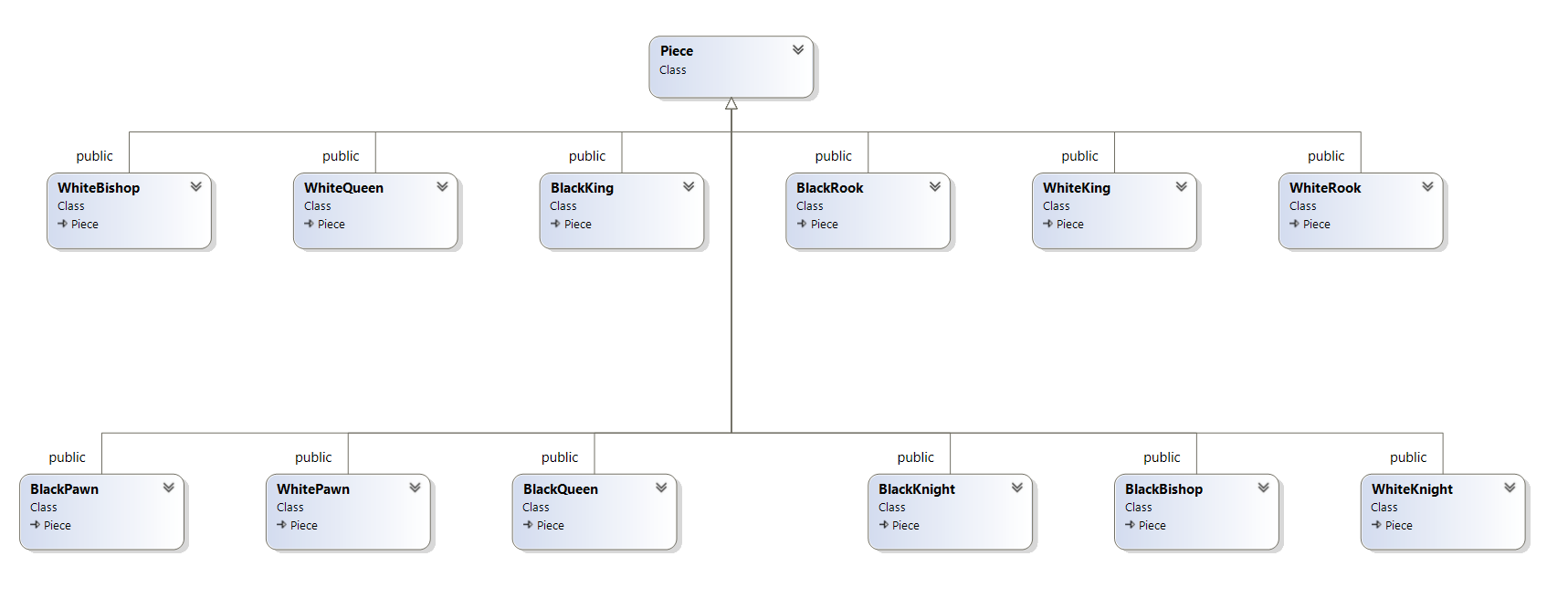
За разработката на играта бе нужен голям набор от цели и задачи, които да се имплементират, за да работи успешно приложението, а именно:

* Изграждане на подходяща архитектура и йерархия на класовете;
* Използване на рационален подход при изграждането на основните функционалности и конструкции на класовете;
* Реализиране на помощни класове с цел подобряване на играта;
* Изработка на подходящ потребителски интерфейс, който съвпада напълно с темата на дадения проект;
* Изграждане на поле за регистрация на играч;
* Изграждане на дъска за игра на шахмат със съответните изисквания за полета,редове и колони;
* Валидация на входа за: Правилно въвеждане на формат за преместване на фигура, а също и инструкции за което.
* Конверитане на входните данни за да могат а бъдат извършвани математически пресмятания за хода който потребителя желае да направи;
* Проверка на всеки един от ходовете спрямо фигурата която ще го извърши (взима се предвид начина по който отделните фигури могат да се местят)
* Възможност за повторно въвеждане на избор при наличието на вече грешен такъв;
* Извеждане на указания и препоръки към потрребителя
* Запазване на играчите и победите в база данни;
* Реализиране на множество помощни функции с цел намаляване на сложността на конкретната задача;

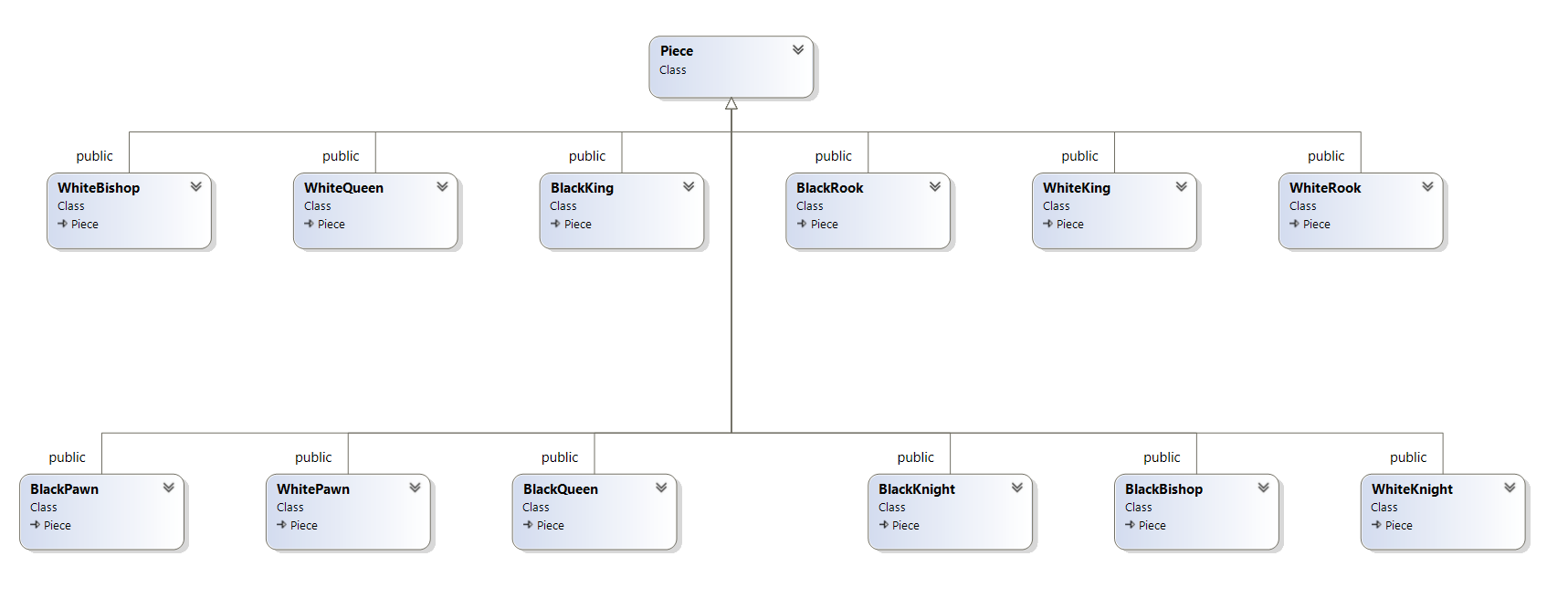
* 1. Структура на документацията:

Първо ще разгледаме архитектурата и йерархията на проекта, по-точно използването на различните свойства на класовете и представянето както на данните им, така и на примитивните им операции. След това ще покажем как операциите се свързват в различните функционалности на играта за нейното успешно реализиране.

1. **Преглед на предметната област:**
   1. . Основни дефиниции, концепции и алгоритми, които ще бъдат използвани:
      1. Клас;
      2. Конструктор, деструктор, копи-конструктор и оператор равно;
      3. Единично наследяване;
      4. Абстрактен клас;
      5. Преобразуване на типове;
      6. Файлове.
   2. Дефиниране на проблеми и сложност на поставената задача:
      1. Създаване на произволна карта с произволни противници;
      2. Четене на играчи от база данни;
      3. Проблеми с конструирането на работещо решение.
   3. Подходи, методи за решаване на проблемите:
      1. Използването на абстрактен клас даде възможността да се направи масив от указатели и чрез преобразуване на типове се постигна искания резултат;
      2. Заместване на имплементацията на база данни от двоичен файл с пряк достъп към текстов файл с последователен достъп.
   4. Потребителски и качествени изисквания:
      1. Поради обхващането на всички случаи чрез валидиране на входа от потребителите не се изисква почти нищо освен при въвеждане на число, да не се въвежда символ.
2. **Проектиране:**
   1. Обща архитектура:
      1. За проекта е използвана съответната архитектура показана на избражението
      2. Базов клас на архитектурата е Piece (Фигура), който съдържа всички общи характеристики а именно:
         * name (име);
         * player (играч)
      3. Производни класове на Този клас са фигурите в шаха, като белите и черните са разделени.



* + 1. Освен това програмата е изградена и чрез класове Board(Дъска) и GamePlay(Ираене).



1. **Рализация**:
   1. Реализиране на класовете:
      * + Класовете са реализирани чрез помощни функции за копиране на необходимите полета с цел оптимизация и съкращаване на код;
        + Класовете са разширени с допълнителни член данни и методи за по-добро реализиране на функционалността на играта;
        + Чрез множество функции за проверки се изклчват невалидните ходове въведени от потребителя и съответно се реализират валидните.
      1. Управление на паметта и реализация на алгоритмите. Оптимизации:
         * Оптимизациите се състоят в използването на допълнителни функции за принтиране, копиране, намиране на дължина на файл и още, запазената дума „break“ в цикли за спиране на излишни проверки, връщане на булеви резултати директно чрез return и още.
      2. Планиране, описание и създаване на тестови сценарии:
         * Интерфейсът е лесен и достъпен, чрез допълнителната информация която се изписва на екрана използването на приложените се улеснява.
         * Навсякъде има пояснителен текст за статуса на играта в текущия момент;
         * Тестов вариант е: Стартиране на играта и реализиране на партия шах
2. **Заключение:**
   1. Обобщение на изпълнението на началните цели:
   2. Крайният продукт е работещо приложение, което отговаря на критериите на ООП и на клиента, като са изпълнени всички функционалности
   3. Насоки в бъдещо развитие:
      1. Кода е написан за лесно негово бъдеще разширение, като може да се добавят нови класове, данни и функционалности без това да застраши изпълнението на предишните такива. Развитието може да се постигне почти във всяко едно отношение, което се дължи на голямата свобода на поставената от клиента задача.

Използвана литература: