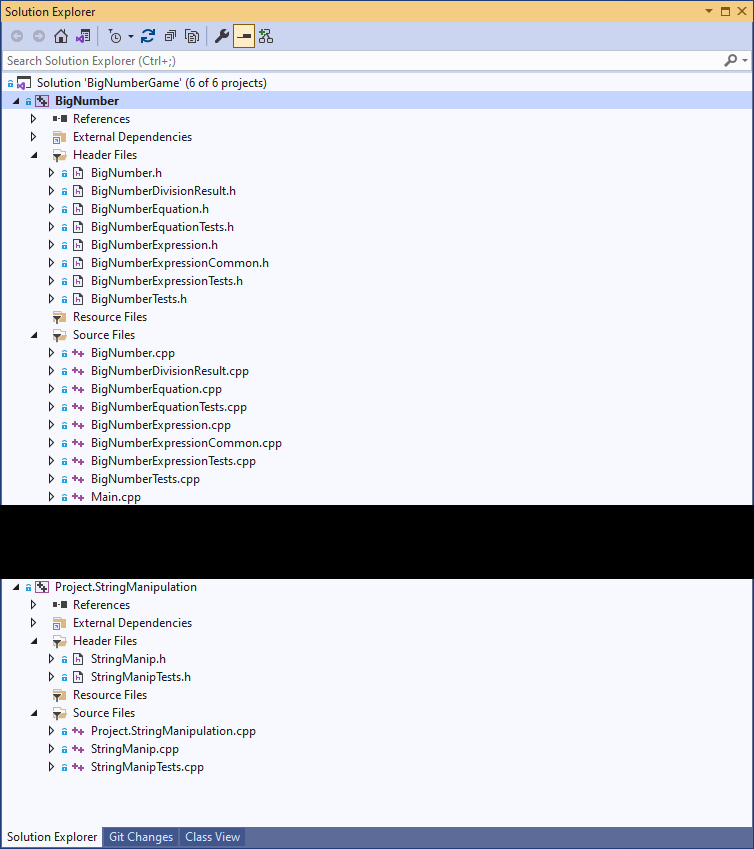
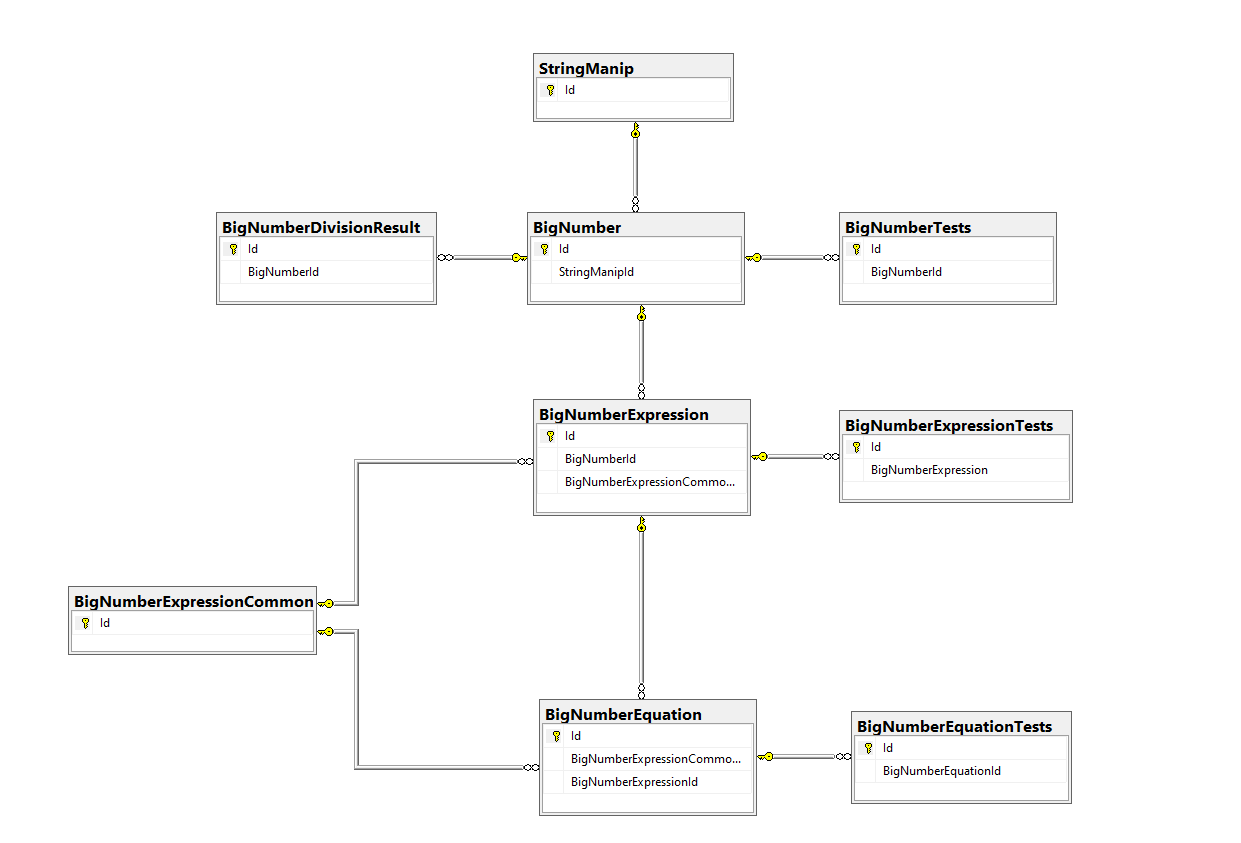
Документация на *BigNumber* проект

1. **Описание и идея на проекта**
   1. Идеята на проекта е реализацията на собствена имплементация на *BigNumber* клас, който представлява целочислен тип данни, който може да съдържа в себе си данни за много по-големи числа, отколкото дори и най-големия стандартен целочислен тип във c++
   2. В последствие реших да разширя целта на проекта, като добавя създаването и на собствени имплементации на класовете *BigNumberExpression* и *BigNumberEquation*, където *BigNumberExpression* съдържа изчислим израз, съставен от *BigNumbers*, а *BigNumberEquation* съдържа уравнение с 0 или повече неизвестни и *BigNumbers*.
   3. В процеса на работа се оказа наложимо да си направя и собствена мини-библиотека за работа със *cstrings* с цел по-лесно боравене с тях и улесняване на работата си по изграждане на функционалността на *BigNumber* и т.н. Тази библиотека е изработена лично от мен на ръка, без копиране на чуждестранен код, така че и тя също влиза в проекта, макар и като по-скоро страничен ефект от създаването на останалите части на проекта, отколкото като конкретно поставена задача.
   4. В проекта влизат и 4 различни тестови класове, които имат за цел да тестват всеки един от съответните класове за тяхната коректна работа.
2. **Основни дефиниции, концепции и алгоритми, които ще бъдат използвани**
   1. Като концепции използвам всичко, взето до 4 май на лекции.
   2. Използва се работа с множество обекти и принципът за ООП.
   3. Използва се основната концепция за *Big 4* при работа с динамична памет. Там, където е нужно, е имплементирана и тествана.
   4. В тази връзка естествено се ползват и познанията за работа с динамична памет и със референции и препратки. Където е необходимо, динамичната памет се освобождава, за да не възникват *memory-leak* проблеми.
   5. Приложена е концепцията за преизползваемост на кода. Тоест, в повечето случаи, където има повтарящ се код, този код се изнася като отделна функция или в един конкретен случай като отделен клас с цел леснота при поддръжка и коригиране на този код в случай на установен проблем в него.
   6. Това се отнася до имплементацията на същинските класове като *BigNumber*, *BigNumberExpression*, *BigNumberEquation* и т.н. Специално изключение правят тестовите класове, където съм пресметнал за   
      по-добре да не изнасям в отделни функции общата функционалност между отделни *test cases* за една и съща функционалност.
3. **Структура на обектите в проекта**
   1. Проектът е написан на Visual Studio. Отдолу са показани отделните компоненти на *BigNumber* проектът.

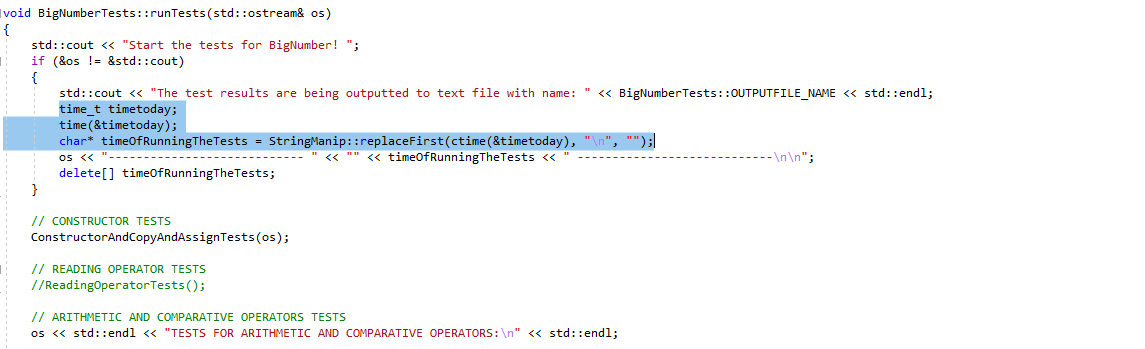


Снимка 1

* 1. Черната част от снимката закрива останалите компоненти от планираният ми до момента втори проект по ООП. Както може да забележите, и целият solution, в който се намира програмата, се казва *BigNumberGame*. Но това за другия път.
  2. За реализацията на този проект съм създал два проекта във Visual Studio.  
     1. *Project.StringManipulation* проектът напрактика е мини-библиотека, съдържаща в себе си помощни функции за работа със null-terminated или още cstrings. Съдържа още и тестовия клас, чрез който се тестват за коректна работа функциите на библиотеката.
     2. *BigNumber* проектът съдържа в себе си всичката функционалност за *BigNumber* и *BigNumberExpression* и *BigNumberEquation*, както и за тяхното тестване.
  3. *Project.StringManipulation* проектът съдържа в себе си:  
     1. Класът *StringManip*, който съдържа в себе си помощните функции за работа със cstrings.
     2. Класът *StringManipTests*, който съдържа в себе си тестове за повечето функции на *StringManip.*
  4. *BigNumber* проектът съдържа в себе си:  
     1. *BigNumber* класът, който в себе си съдържа цялата функционалност за работа с големи числа.
     2. *BigNumberDivisionResult* класът, който съдържа информацията за частното и остатъка от някакво деление на големи числа.
     3. *BigNumberTests* класът, който съдържа в себе си всички тестове за *BigNumber.*
     4. *BigNumberExpression* класът, който съдържа функционалността за изрази, съставени от големи числа, в себе си.
     5. *BigNumberExpressionTests* класът, който в себе си съдържа тестовете за *BigNumberExpression.*
     6. *BigNumberEquation* класът, който в себе си съдържа функционалността за уравнения, съставени от големи числа.
     7. *BigNumberEquationTests* класът, който в себе си съдържа тестовете за *BigNumberEquation.*
     8. *BigNumberExpressionCommon* класът, който съдържа в себе си виртуални функции, необходими за генерирането на случаен принцип на компоненти от изразите и уравненията. *BigNumberExpression* и *BigNumberEquation* класовете го наследяват.
  5.  Диаграма на зависимостите на различните компоненти на проекта:

Диаграма 1

* 1. Зависимостите, казани с думи:  
     1. *BigNumber* е директно зависим от *StringManip.*
     2. *BigNumberDivisionResult*, *BigNumberTests* и *BigNumberExpression* са директно зависими от BigNumber.
     3. *BigNumberExpression* е директно зависим от *BigNumberExpressionCommon.*
     4. *BigNumberExpressionTests* и *BigNumberEquation* са директно зависими от BigNumberExpression.
     5. *BigNumberEquation* е директно зависим от *BigNumberExpressionCommon.*
     6. *BigNumberEquationTests* е директно зависим от *BigNumberEquation.*

1. **Реализация и тестове**
   1. Погледнете в *Doxygen* генерираната документация.
   2. В сорс кодът на проекта.
2. **GitHub**
   1. <https://github.com/rosenkolev1/University-of-Sofia-cplusplus-exercises>   
      Това е цялото ми *repo*, което съдържа всички неща, свързани с програмирането в университета, включително и BigNumber проекта.
   2. <https://github.com/rosenkolev1/University-of-Sofia-cplusplus-exercises/tree/main/Computer%20Science%20Year-1%20Term-2/BigNumberGame>  
      Това е директорията на *solution-на*, в който се намира и проекта. Същият *solution* е показан и на „снимка 1“.
3. **Източници**
   1. Типичните сайтове, които излизат в търсачката на гугъл, когато попитаме как се прави нещо на c++: *geeksforgeeks.com*, *cppreference.com*, *stackoverflow.com*, в някои случай, понеже пиша на visual studio, и майкрософтската документация, и т.н.
   2. Чужд код обаче съм копирал точно един път. Ето го и него:
   3. Кодът има за цел да изпише текущото време на стартиране на тестовете. Това не знаех как става в c++, за това погледнах в интернет и частта с *time\_t* от *stackoverflow*.
   4. Дори и за използването на *srand* и *rand* не ми трябваше да копирам от интернет, понеже вече имах от преди това програма, в която се използваше. Мисля че беше от второто домашно от първия семестър.

*Изработил: Росен Андреев Колев, 4-та група, 0MI0800065*