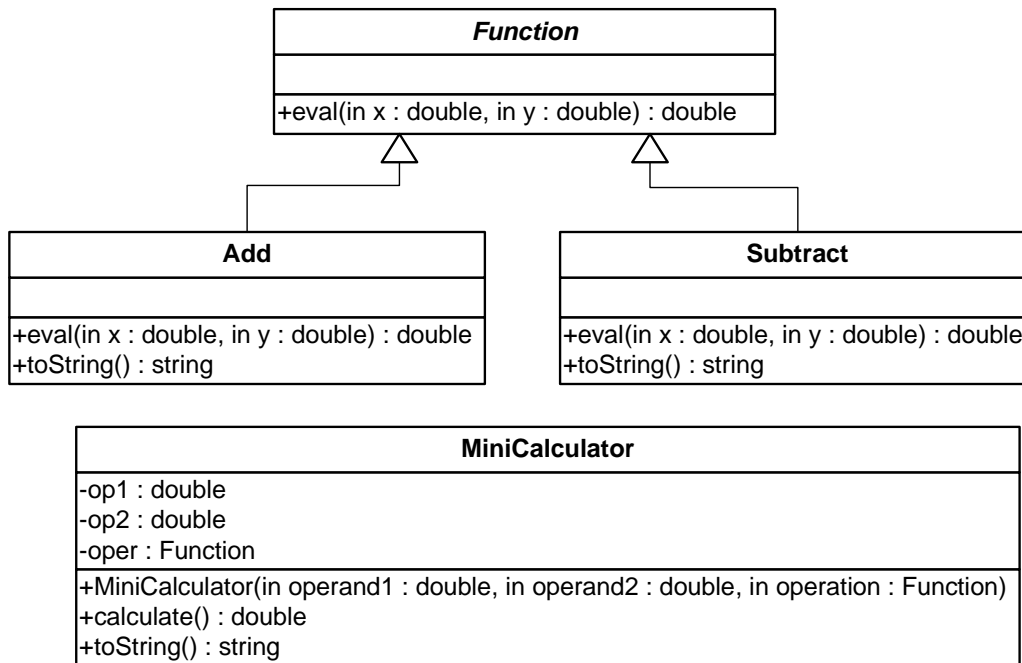


Упражнение №5

Абстрактни класове

I. Йерархия на класове. Полиморфизъм чрез абстрактен клас.



1. Декларирайте абстрактен клас **Function** (**Функция**) с един абстрактен метод **eval**, който изпълнява произволна операция с два **double** аргумента и връща резултат от тип **double**.
2. Декларирайте клас **Add** (**Събиране**), който наследява абстрактния клас **Function**, реализира операцията **събиране** на два операнда и предефинира метода **toString**, така че да изобразява операцията събиране (" + ").
3. Декларирайте клас **Subtract** (**Изваждане**), който наследява абстрактния клас **Function**, реализира операцията **изваждане** на два операнда и предефинира метода **toString**, така че да изобразява операцията изваждане (" - ").
4. Декларирайте клас **MiniCalculator** (**Мини калкулатор**) с три полета: **ляв операнд** от тип **double**, **десен операнд** от тип **double** и **операция** от тип **Function**.
5. Добавете конструктор с три параметъра съответно от тип **double**, **double** и **Function** за инициализиране на ляв операнд, десен операнд и операция.
6. Добавете метод **calculate**, който извършва операцията, използвайки полиморфно извикване на метода **eval** чрез референция към операцията на абстрактния клас **Function** и връща резултат от тип **double**.
7. Предефинирайте метода **toString**, за да изобрази изчисления израз.
8. За тестване на класовете декларирайте клас **TestMiniCalculator** и в статичния метод **main** напишете следния код:
 - а) декларирайте масив **calc** от типа на класа **MiniCalculator** и отделете памет за две операции;
 - б) създайте две операции: **събиране** с операнди 5 и 6 и **изваждане** с операнди 2 и 3;
 - в) изведете на екрана изчислените изрази.
9. Компилирайте и изпълнете приложението. Методът **MiniCalculator.toString** се обръща полиморфно към методите **toString** и **calculate**, като последният извиква полиморфно метода **eval**.

10. Декларирайте още два класа **Multiply** (**Умножение**) и **Divide** (**Деление**), които наследяват абстрактния клас **Function**, реализират операциите **умножение** и **деление** на два операнда, и предефинират метода **toString**, така че да изобразява съответната операция.
11. Модифицирайте тествания клас **TestMiniCalculator**, така че да демонстрирате изчислението на четирите вида операции: събиране, изваждане, умножение и деление на две реални числа.
12. Компилирайте и изпълнете приложението.

II. Задачи за самостоятелна работа

1. Създайте абстрактен клас **AbstractConvert** (**Абстрактен конвертор**), който съдържа:
 - а) поле за изходната стойност **fromValue** от тип **double** с **protected** модификатор за достъп (наследниците ще имат достъп до полето);
 - б) конструктор с **protected** модификатор на достъп и един параметър, който инициализира полето за изходната стойност (абстрактният клас не може да създаде инстанция, но конструкторът му е с **protected** модификатор за достъп, тъй като неговите наследници ще го използват);
 - в) абстрактен метод **convert**, връщаш преобразуваната стойност от тип **double**.
2. Създайте подкласове на **AbstractConvert**:
 - а) клас **FahrenheitToCelsius** (**Фаренхайт в Целзий**). Добавете конструктор с един параметър. Реализирайте метода **convert**, който преобразува от градуси по Фаренхайт в градуси по Целзий. Предефинирайте метода **toString**, така че да изобразява подходящо съобщение за изходната и преобразуваната стойност;
 - б) клас **CelsiusToFahrenheit** (**Целзий във Фаренхайт**). Добавете конструктор с един параметър. Реализирайте метода **convert**, който преобразува от градуси по Целзий в градуси по Фаренхайт. Предефинирайте метода **toString**, така че да изобразява подходящо съобщение за изходната и преобразуваната стойност;
 - в) клас **LitersToGallons** (**Литри в галони**). Добавете конструктор с един параметър. Реализирайте метода **convert**, който преобразува от литри в галони. Предефинирайте метода **toString**, така че да изобразява подходящо съобщение за изходната и преобразуваната стойност;
3. Създайте тестов клас, в който дефинирайте масив **conv** от абстрактния клас **AbstractConvert**, инициализирайте елементите му с различен тип клас-наследник и изведете на екрана информация за изходните и преобразуваните стойности.

