

Формули

Част първа: Йерархия от класове (5 т.)

Да се дефинира абстрактен базов клас `Formula`, дефиниращ операцията `double value() const`. Класът да представя формула с неуточно представяне, чиято стойност може да бъде изчислена чрез метода `value`. Да се дефинира също и операция `void print() const` за извеждане на явния вид на формула на стандартния изход.

а) Да се дефинира производен клас `Constant`, който описва дробни константи. Обекти от клас `Constant` се конструират със стойността на съответната константа.

Пример:

```
Constant a (0), b (1);
std::cout << b.value();    // извежда 1
b.print();                 // също извежда 1
```

б) Да се дефинира клас `BinaryOperation`, описващ приложение на двуместна аритметична операция върху други две формули. Допустимите операции са `+`, `-`, `*`, `/`, както и операциите за сравнение `<` и `=`, като последните имат стойност 1 при удовлетворено (не)равенство и стойност 0 иначе. Обекти от клас `BinaryOperation` се конструират със символ, описващ операцията и два указателя към формули, задаващи операндите.

Пример:

```
BinaryOperation test ('*',
    new Constant (2),
    new BinaryOperation ('+',
        new Constant (1),
        new Constant (3));
std::cout << test.value();
// извежда 8, т.е. стойността на 2 * (1 + 3)
test.print();    // извежда "( 2 * ( 1 + 3 ) )"
```

в) Да се дефинира клас `Read`, описващ стойност, въведена от потребителя.

Пример:

```
BinaryOperation test ('*', new Constant (2), new Read);
std::cout << test.value(); // изисква от потребителя да въведе
                          // число и го връща умножено по 2
test.print();            // извежда "( 2 * Read )"
```

г) Да се дефинира клас `Conditional`, описващ формула, която зависи от дадено условие. Обекти от клас `Conditional` се конструират с три указателя към `Formula` — условие ("if"), формула при удовлетворяване на условието ("then") и формула при неудовлетворяване на условието ("else").

Пример:

```
Conditional test (new BinaryOperation ('<', new Read,
                                          new Constant (5)),
    new Constant (1),
    new Constant (2));
std::cout << test.value(); // извежда 1, ако поребителят въведе
                          // число, по-малко от 5
                          // и 2 в противен случай
test.print(); // извежда "if ( Read < 5 ) then 1 else 2"
```

Забележка: за тази част от задачата не се изискват никакви усилия по управление на динамичната памет.

Втора част: Управление на динамичната памет (3 т.)

За така изградената йерархия да се дефинират необходимите специални методи (конструктори, деструктори, оператор за присвояване) така, че динамичната памет да се управлява коректно. Следните операции трябва да могат да протичат без проблеми, свързвани със споделяне на памет, изтичане на памет или други неправилни манипулации на динамичната памет. След приключването на съответния им блок, заетата динамична памет трябва да бъде освободена.

```
Constant c1 (1);
Constant* c2 = new Constant (2);
BinaryOperation f1 ('+', &c1, c2);
BinaryOperation f2 = f1;      // копиране
f2 = f1;                      // присвояване
delete c2;
```

Упътване 1: В класа `Formula` да се добави виртуален метод `Formula* clone()` `const`. Всеки клас да предефинира метода така, че обектите да могат да построят свои идентични копия.

Упътване 2: Да се променят класовете така `BinaryOperation` и `Conditional`, така че да “притежават” формулите, които ги съставят. В частност конструкторите да бъдат построени така, че да запомнят в член-данните адресите на копия на параметрите си, а не оригинално подадените адреси. Деструкторите, от своя страна, да изтриват копията, записани в съответните член данни.

Трета част: Десериализация (4 т.)

Показаните в примерите резултати от метода `print` за всеки клас от описаната йерархия демонстрират едно еднозначно представяне на всички формули, представими с нея. Да се дефинира рекурсивна функция `Formula* readFormula()`, която по въведено от стандартния вход правилно построено представяне на формула построява съответното ѝ представяне чрез класовете от изградената йерархия.

Пример:

```
Formula* f = readFormula(); //от стандартния вход се въвежда
                             // "if ( Read < 5 ) then 1 else 2"
std::cout << f->value(); // извежда 1, ако поребителят въведе
                        // число, по-малко от 5
                        // и 2 в противен случай
delete f;
```

Забележка: По ваше желание можете да изберете друго, по-просто представяне на формула.