

Пойнтъри, референции и функции 3 в 1

- Да повторим какво става ако имаме референция като параметър
- Какво става, ако имаме пойнтър като параметър
- Какво става, ако имаме референция към пойнтър като параметър
- Какво става, ако функцията връща референция
- Какво става, ако функцията връща пойнтър
- Някои добри практики, с които ще е по-трудно да се гръмнете в крака по невнимание

Референция като параметър

- Създава се нов обект, чиито данни са на адреса на формален параметър
- Няма копиране на данни
- Тъй като новият обект е пряко свързан с оригиналния, то каквито и промени да му направим, ние променяме и оригиналния

Пойнтър като параметър

- Създава се нов обект, който копира информацията на формалния параметър
- Този нов обект сочи към същия адрес като оригиналния
- Ако извършим промени в адреса, към който сочи новият обект, то ние ще променим информацията там (същата информация, към която сочи и оригиналният пойнтер)
- Ако сменим адреса, към който сочи новият пойнтер, няма да сменим адреса, към който сочи оригиналният пойнтер, защото двата обекта са различни и не са пряко свързани

Референция към поинтър като параметър

- Поинтърът е обект, който си има адрес и стойност => няма причина да няма референция към поинтър
- Синтаксис <тип> * & <име> = <lvalue>
- Аналогично както при всички референции, щом се създаде референция към поинтър, то вече има 2 начина да се обърнем към един и същи обект
- Ако променим към какво сочи поинтърът през което и да е име, променяме към какво сочи самият обект
- Ако имаме референция към поинтър като формален параметър, то важат абсолютно същите правила, за които говорихме досега

Функция, връщаща референция

- Доста tricky елемент за УП
- Много е лесно да се простреляте в крака на този етап, но ще ви го покажа, защото в бъдеще ще ви е полезно
- Когато връщате референция, вие не връщате стойността на променливата, а цялата променлива
- Трябва да сте сигурни, че променливата, чиято референция връщате, съществува и след приключването на функцията, тоест не връщате локално създаден обект
- [Demo3](#)

Функция, връщаща референция

- Пример за грешна функция връщаща референция е

```
int & errorProne()  
{  
    int a = 5;  
    return a;  
}
```

Недефинирано поведение, което компилаторът на Visual Studio, любезно заличава, но реално това е проблем и не всички компилатори го позволяват

Функция, връщаща поинтър

- Поинтърът е най-обикновен обект => когато една функция връща поинтър, тя връща нов обект, който има за стойност адреса, към който сочи оригиналният поинтър
- Щом се връща нов обект, то той не е свързан с оригиналния, но въпреки това сочат към един и същи адрес и ако се извърши дереференциране (влизане в клетката), то ще бъде променена информацията в адреса, който съдържат и двата поинтъра

Някои добри практики, с които ще е по-трудно да се гръмнете в крака по невнимание

- Можете да връщате референции и пойнтери към константи, както и да подавате такива формални параметри във функция
- Винаги създавайте константи, когато нямате намерение да промените обекта, липсата на 1 константа може да ви коства много време и главоболия!
- Референциите към примитивни типове данни и пойнтери не спестяват толкова много време и памет колкото си мислите, затова не е добра практика да ги използвате, освен ако не искате да промените оригиналните обекти