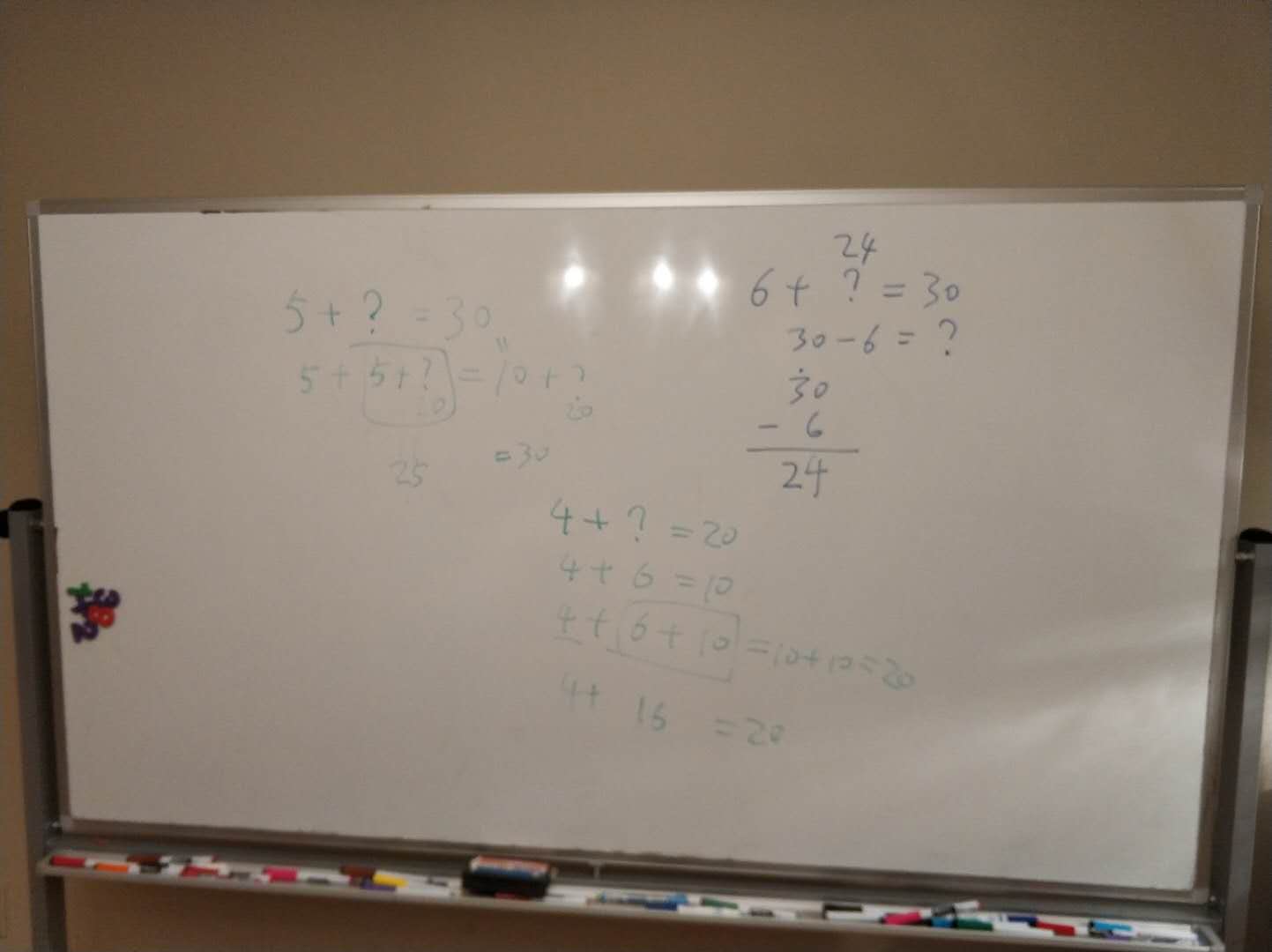
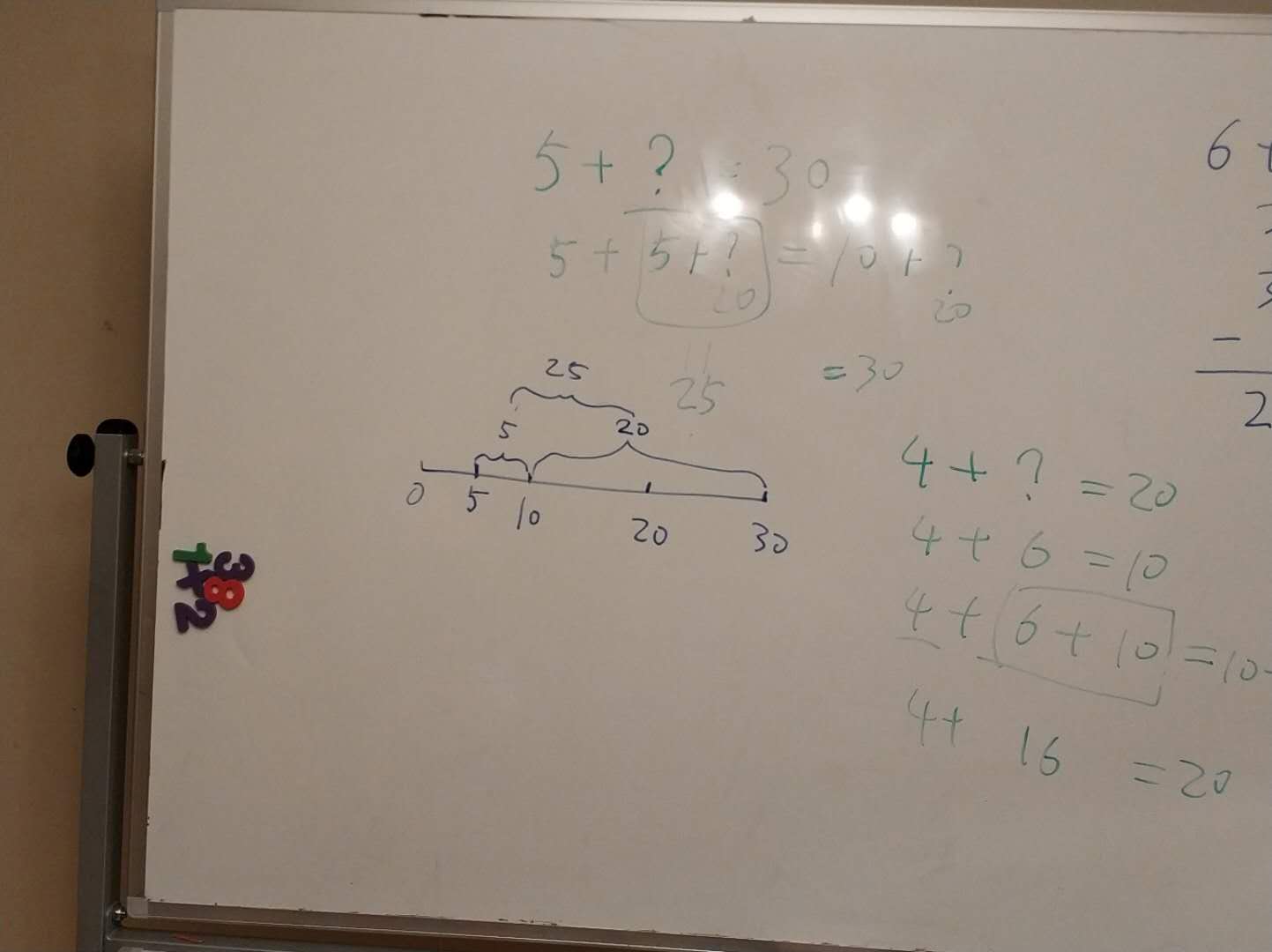
1月19日，启蒙数学小小班

这次的重点是练习凑整数。练习的方式是一个小朋友说一个小于等于10的数，下一个小朋友说出另一个数使得两数相加等于10。例如小朋友甲说3，小朋友乙就说7，使得3+7=10。然后把10换成20、30、乃至更高。

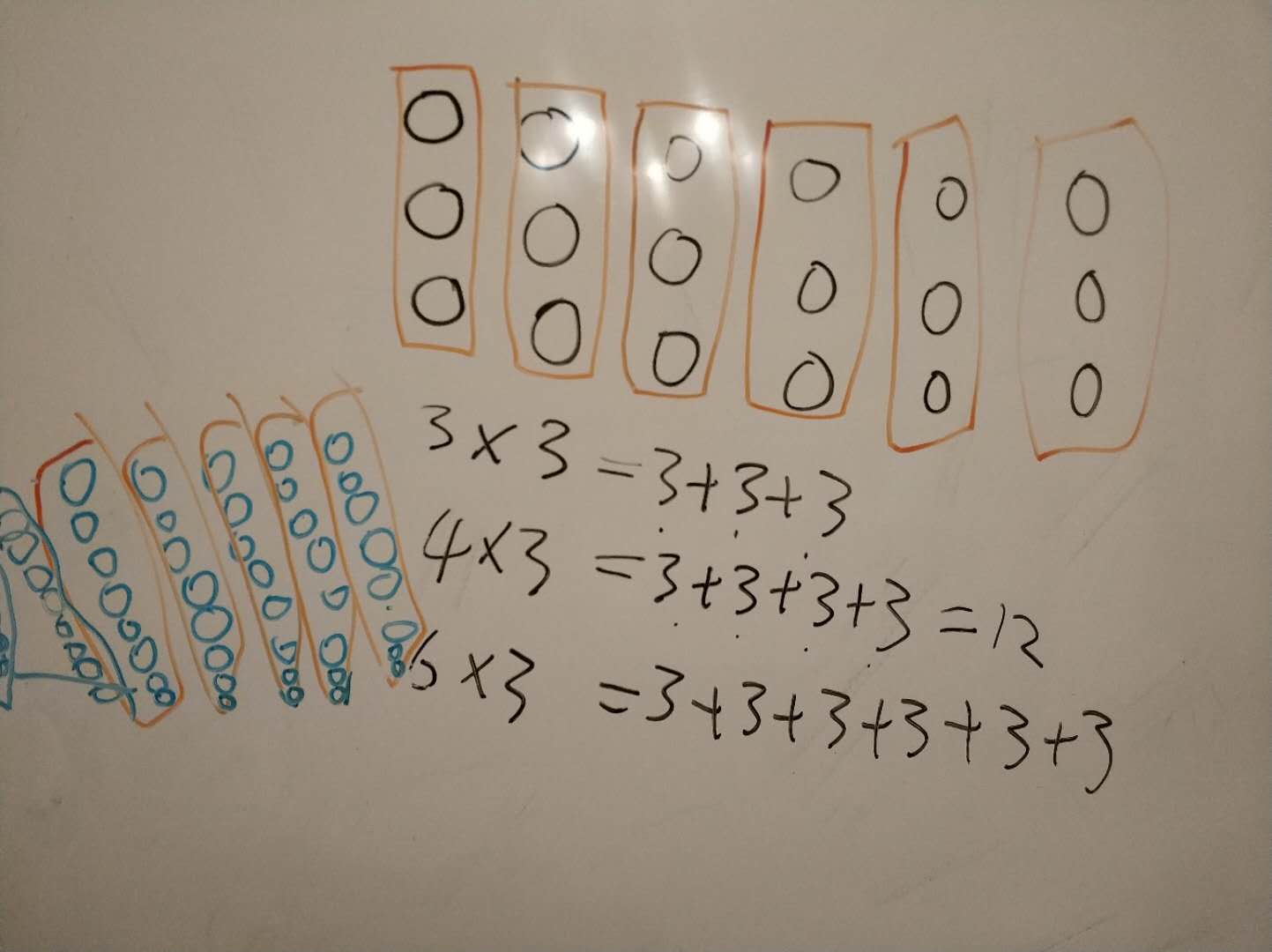


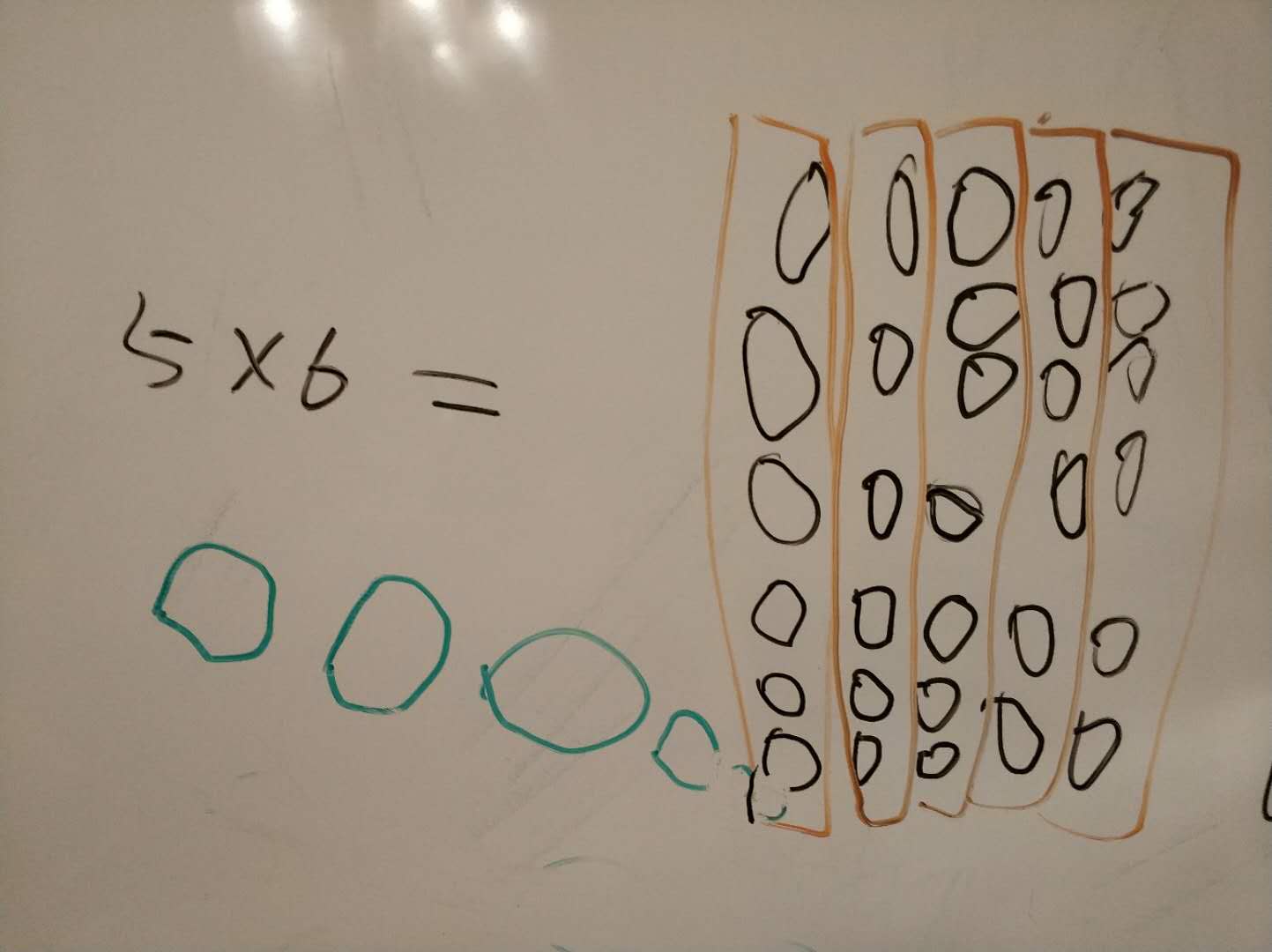


这个练习的目的是为小朋友们的快速心算夯实基础。凑10的时候小朋友们都算得很快，凑20的时候，因为涉及到个位数相加进位的问题，小朋友们就慢下来了。这是因为计算变成了两步走：第一步凑10，第二步凑20。例如：4 + ? = 20，第一步是4 + 6 = 10，第二步是10 + 10 = 20，综合起来就是4 + 6 + 10 = 20 –> 4 + (6 + 10) = 4 + 16 = 20。

凑30的计算与此类似：5 + ？ = 30，第一步是5 + 5 = 10，第二步是10 + 20 = 30。所以5 + （5 + 20） = 5 + 25 = 30。这种两步走的算法因为既涉及个位数的凑整也涉及十位数的加法，除了极少数小朋友，大多数小朋友们都容易卡壳——我个人猜测是小朋友们的大脑发育还没到能够同时处理多个步骤的地步（编程的黑话就是“内存不够”）。

接下来讲乘法的本质：乘法的本质是加法的一种速记符号。例如4 3是3 + 3 + 3 + 3的速记符号，而九九乘法表则是对一些已知加法结果的汇总，以记忆的方式固化在使用者脑中，便于快速查询。所以九九乘法表和对数表、正态分布表等数学表格没有本质区别。





最后讨论一下强化练习对数学学习的意义，或者更广泛的意义上，“守规矩”和“创新”的关系。

中国小学生一二年级练习速算，主要就是训练熟练程度，比如7 + 8 = 15要达到条件反射的地步。我个人认为这是合理的也是必要的。这种对基本规则的熟练使用，即使在大学的微积分教学中，也是必要的训练。这和“死记硬背”不太一样，更接近于“不逾规矩而随心所欲”。 “创新”通常来自于对事物本质的深刻理解（比如相对论、群论等理论研究），而深刻理解则来自于反复揣摩，而反复揣摩的必要条件就是反复练习——当然，这里涉及到练习题的设计水平问题，有兴趣者可以搜索北大的张筑生教授当年为中国国际奥数参赛队设计训练题的故事。<https://www.guancha.cn/Education/2017_05_26_410230_s.shtml>：“当奥赛教练最难的是出题，因为参赛选手都是中学生，所以每道题都要把高等数学初等化，这种倒推的过程是非常艰苦的。要搞出一个既有新意，又有难度，还能分出学生优劣的题目，至少需要一周的时间，而张筑生每年都能拿出几十道新题出来。”

学习数学到底有什么用？作为半个软件工程师，我觉得抽象代数的学习对于软件架构的设计至关重要，因为好的软件架构需要对功能进行适当的抽象，使得软件既便于编写维护又便于拓展升级。计算机科学中的圣经，《计算机编程的艺术》（The Art of Computer Programming）从本质上讲就是一本数学书。作者Donald Knuth的算法设计充分体现了抽象性对软件设计的重要性。

抽象化思考和实际动手能力的关系，有点类似于武侠小说里的内功和剑法拳脚的关系。中国有句老话：“练拳不练功，到老一场空；练功不练拳，犹如无舵船”。抽象化思考帮助我们提炼问题的本质，而动手能力则把这种提炼转化成实际的科研成果或者商业产品。如蒙氏教育和林老师使用的Monster School教材那样，用种种直观方式讲述加法进位的本质，是练内功——“理解客观世界”，但要真正出活，“改造客观世界”，就需要练拳做练习题了。下面是林老师这次布置的家庭作业练习题。

