



Liberté • Égalité • Fraternité

RÉPUBLIQUE FRANÇAISE

BI3XM19V1

CONCOURS EXTERNE ET INTERNE DES 4, 5 ET 6 FÉVRIER 2019
POUR LE RECRUTEMENT DE CONTRÔLEURS DES DOUANES ET DROITS INDIRECTS
BRANCHE DU CONTRÔLE DES OPÉRATIONS COMMERCIALES ET D'ADMINISTRATION GÉNÉRALE
SPÉCIALITÉ TRAITEMENT AUTOMATISÉ DE L'INFORMATION – PROGRAMMEUR

Zones 1&2

ÉPREUVE ÉCRITE D'ADMISSIBILITÉ N° 3

(DURÉE : 1 HEURE 30 – COEFFICIENT 1)

**TRADUCTION D'UN TEXTE EN ANGLAIS ISSU D'UNE REVUE
OU D'UNE DOCUMENTATION INFORMATIQUE**

AVERTISSEMENTS IMPORTANTS

L'usage du dictionnaire et de tout document autre que le support fourni est **interdit**. Seul le matériel usuel d'écriture est autorisé.

Toute fraude ou tentative de fraude constatée par la commission de surveillance **entraînera l'exclusion du concours**.

Veillez à bien indiquer sur votre copie le nombre d'intercalaires utilisés (la copie double n'est pas décomptée).

Il est interdit de quitter définitivement la salle d'examen avant le terme de la première heure.

Le présent document comporte **2 pages** numérotées.

Tournez la page, SVP

Pair programming and how it can make you a better programmer

Pair programming, where two developers share a single workstation, has grown more common and less controversial over the years, but many still haven't tried it themselves. In pair programming one participant is the "driver," who actually writes code, and the other is the "navigator," who checks the driver's work as it's done and keeps an eye on the big picture. The two switch roles every few minutes, and talk to each other throughout the process. (...)

Two programmers of differing skill levels

When a master and a novice are paired, the novice is in an optimal position to learn and grow very quickly. When the master is the driver, the novice is observing not just what code is being produced, but how it's being produced. The process gives insight into the thought process, shortcuts, and style of the master developer. It's akin to watching a gourmet chef cook as opposed to reading a recipe. While the driver narrates as he codes, the novice is able to ask questions and engage in discussion in the moment.

When it's time for the novice to drive, he has the oversight and guidance of the more experienced developer beside him. He can make mistakes without fear, and use them as learning moments.

The benefits to the novice are clear, but the master programmer also benefits from the experience. He is exposed to new ideas and thoughts, and made to think critically about his choices so he can explain them clearly. This also cuts down on time put aside specifically for training and teaching that takes away from work; the pair can be doing real work together as they teach and learn.

Two programmers of similar skill levels

When two developers of similar skill levels pair program, there are other benefits. This technique is used effectively to onboard new developers, for example. While their technical skills may be equally matched, one has much more institutional knowledge that can be transferred to the other while accomplishing work together as a pair.

When two programmers with similar skill levels and similar levels of institutional knowledge are paired as part of daily operations, the teaching and learning effect can continue, and the final product will likely be of higher quality than if developers had worked independently. When a developer is coding, it's easy to get lost in the weeds and the details of the problem. If there's a navigator by your side, he can take a bigger-picture view of the work, bouncing ideas around and pulling your head up for a moment to consider alternative ways to solve a problem.

(...) Wherever you are in your career, there is always somebody who knows things you don't. In fact, everybody knows things you don't. And you know things others don't. Collaboration has always been valued as a way for skilled workers to share their experience, technique, and ideas; programming is no exception.

(...)

Alyssa Mazina, StackOverflow Blog, 22/11/16