

Programátor || Programátorka

Náplň práce

- Vývoj zásuvných modulů do systému DokuWiki, jejich integrace do existujícího systému.
- rozvoj existujících PHP aplikací,
- správa systému DokuWiki a souvisejících MySQL databází,
- komunikace s vedoucími pracovníky, sběr podkladů, návrh řešení, implementace řešení,
- tvorba dokumentace.

Nutné podmínky

- Pokročilá znalost PHP, MySQL, HTML a CSS,
- znalost anglického jazyka,
- flexibilita, ochota se učit nové věci, komunikativnost, proaktivita.

Ideálně také

- Zkušenosti s administrací některého z open-source redakčních systémů DokuWiki, Drupal, WordPress,
- zkušenost s vývojem webových aplikací pod PHP-MySQL,
- znalost JavaScriptu a DokuWiki.

Nabízíme

Práci na DPP,

- zajímavé platové ohodnocení s možností navýšení po zapracování,
- práci v Brně, možnost práce z domova,
- předpokládaný začátek spolupráce únor či březen 2019,
- navazující dlouhodobou spolupráci formou DPČ.

Životopis a motivační dopis (rozsah max. 1 strana A4) nám odesílejte elektronicky pomocí níže uvedeného formuláře do **27. 1. 2019**. Odesílání přihlášek uzavřeno.

Not just a coder?

Specification

```
class Recruiter implements HappyEmployeeInterface
{
    const COM_MODE_PERSONAL = 1;
    const COM_MODE_PHONE = 2;
    const COM_MODE_SMS = 3;
    const COM_MODE_EMAIL = 4;
    const COM_MODE_NONE = 5;

    private $employee;
    private $vis;
    private $wage;
    private $month_limit;
    private $year_limit;
    private $hour_sum;
    private $contract_type;

    function __construct(Person $emp,
                          VidaInfoSystem $vis,
                          $contract_type =
VidaInfoSystem::CONTRACT_TYPE_DPP,
                          $wage = 0xE6,
                          $month_limit = 10000/230,
                          $year_limit = 300)
    {
        $this->employee = new VidaEmployee($emp);
        $this->vis = $vis;
        $this->wage = $wage;
        if ($contract_type ==
VidaInfoSystem::CONTRACT_TYPE_DPC)
        {
            if (is_null($month_limit)) $month_limit = 0;
            if (is_null($year_limit)) $year_limit = 1040;
        }
        $this->month_limit = $month_limit;
        $this->year_limit = $year_limit;
        $this->contract_type = $contract_type;
        $this->hour_sum = $this->vis->get_current_hour_sum();
    }

    public function communicate($info, $mode =
self::COM_MODE_PERSONAL)
    {
        switch ($mode)
        {
            case self::COM_MODE_PERSONAL:
                return $this->vis->solution($info,
VidaInfoSystem::EFFECTIVITY_BEST);
            case self::COM_MODE_PHONE:
                return $this->vis->solution($info,
VidaInfoSystem::EFFECTIVITY_OK);
            case self::COM_MODE_SMS:
                return $this->vis->solution($info,
VidaInfoSystem::EFFECTIVITY_OK);
            case self::COM_MODE_EMAIL:
                return $this->vis->solution($info,
VidaInfoSystem::EFFECTIVITY_OK);
            case self::COM_MODE_NONE:
                return
$this->vis->fire_employee($this->employee);
        }
        return $this->error('Unacceptable communication mode
' . $mode);
    }

    public function workflow()
    {
        while ($this->employee->get_time_status() ==
VidaEmployee::HAS_FREE_TIME)
        {
            $spec = $this->vis->get_task_specification();
            while ($spec->status !=
VidaInfoSystem::TASK_STATUS_FINISHED)
            {
                $status = $this->employee->solve_task($spec);
                $status = $this->communicate($status,
$this->employee->get_communication_mode());
            }
            return $this->employee->satisfaction_level();
        }
    }

    private function payment($hours)
    {
        $this->vis->previous_accumulation($hours);
        if ($hours + $this->hour_sum > $this->year_limit)
        {
            msg('You are a "STACHANOVEC"');
            return $this->error('Year contract hour limit can
not be exceeded!');
        }
        if ($hours > $this->month_limit)
        {
            $this->vis->negotiate_accumulation_of_hours2next_month($hours -
$this->month_limit);
            $hours = $this->month_limit;
            msg('Partial transfer negotiated');
            $this->warning('Month contract hour limit can not
be exceeded!');
        }
        if ($hours < 15)
        {
            $this->vis->negotiate_accumulation_of_hours2next_month($hours);
            msg('Transfer negotiated. ');
            return 0;
        }
        $this->hour_sum += $hours;
        return $this->wage * $hours;
    }

    private function evaluation()
    {
        $eval =
$this->vis->get_employee_evaluation($this->employee);
        if ($eval > VidaEmployee::EVALUATION_OK)
        {
            $this->wage *= hex2float('3ff921fb54442d18');
        }
    }

    // https://stackoverflow.com/questions/34067202
    private function hex2float($strHex)
    {
        $hex = sscanf($strHex,
"%02x%02x%02x%02x%02x%02x%02x");
        $hex = array_reverse($hex);
        $bin = implode('', array_map('chr', $hex));
        $array = unpack('dnum', $bin);
        return $array['num'];
    }
}
```

From:

<https://www.mscb.cz/> - **MSCB**

Permanent link:

<https://www.mscb.cz/cz/prace/programator2>

Last update: **2019/01/17 17:37**

