

Programista

na rozmowie kwalifikacyjnej



IF

knowledge

+ skills

+ preparation

+ understanding_recruiter

THEN

good_performance

new_job

great_career



Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiegokolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie książki na nośniku filmowym, magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi bądź towarowymi ich właścicieli.

Autor oraz Wydawnictwo HELION dołożyli wszelkich starań, by zawarte w tej książce informacje były kompletne i rzetelne. Nie biorą jednak żadnej odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub autorskich. Autor oraz Wydawnictwo HELION nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Opieka redakcyjna: Ewelina Burska

Projekt okładki: Studio Gravite/Olsztyn

Obarek, Pokoński, Pazdrijowski, Zaprucki

Materiały graficzne na okładce zostały wykorzystane za zgodą Shutterstock.

Wydawnictwo HELION

ul. Kościuszki 1c, 44-100 GLIWICE

tel. 32 231 22 19, 32 230 98 63

e-mail: helion@helion.pl

WWW: <http://helion.pl> (księgarnia internetowa, katalog książek)

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

<http://helion.pl/user/opinie/prorkw>

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

ISBN: 978-83-283-2399-5

Copyright © Helion 2019

Printed in Poland.

- [Kup książkę](#)
- [Poleć książkę](#)
- [Oceń książkę](#)

- [Księgarnia internetowa](#)
- [Lubię to! » Nasza społeczność](#)

Spis treści

Wstęp	27
--------------------	-----------

Część I	29
----------------------	-----------

Rozdział 1. Po co, jak często i czy w ogóle chodzić na rozmowy rekrutacyjne?	31
---	-----------

Programista bez rozmów?	31
Chodzić, nie chodzić?	32
Jak często chodzić na rozmowy?	33
Udawanie odejścia, by „wrócić”	35
Czy skoczki zarabiają więcej?	36

Rozdział 2. Wybór firmy, wybór stanowiska	41
--	-----------

Polska czy zagraniczna?	42
Duża czy mała?	43
Informatyczna czy nieinformatyczna?	44
Nowa na rynku czy już z historią?	45
Jaka technologia, jaka warstwa?	47
Etat czy kontrakt?	48

Rozdział 3. CV	51
-----------------------------	-----------

Przedwcześnie ogłaszana śmierć CV	51
Cztery cechy dobrego CV i bonus	53
1. <i>Kompletność</i>	53
2. <i>Komunikatywność</i>	54
3. <i>Przejrzystość</i>	55
4. <i>Zwięzłość</i>	56
5. <i>Otwieralność</i>	57
Przydatne i zbędne elementy CV	57
Gdy nie masz czym wypełnić CV	60

Rozdział 4. Jak współpracować z rekruterami	63
Rodzaje rekruterów	63
Jak rozpoznać dobrego rekrutera?	64
Jak możesz pomóc rekruterowi?	67
Jak rekruter może pomóc Tobie?	68
Za co można trafić na czarną listę u rekrutera?	69
Rozdział 5. Przygotowanie do rozmowy	73
Wstępna rozmowa telefoniczna	73
Kiedy rozmawiasz z... aplikacją	74
Sprawdź, gdzie idziesz, czyli research na temat firmy	76
Sprawdź, co jest na Twój temat w sieci, czyli research na temat siebie	77
CV i opis stanowiska — dwa dokumenty do przeczytania	78
Logistyka i inne praktykalia	79
Rozdział 6. Kluczowe 60 minut	81
Wygląd	81
Rekrutacyjny savoir-vivre	83
Jak mówić, by zjednać sobie rekrutera?	85
O co powinieneś spytać na rozmowie (a o co lepiej nie)?	87
Referencje	89
Rozmowa kwalifikacyjna w pubie	90
Po spotkaniu	91
Rozdział 7. Łatwe, trudne, głupie — o pytaniach rekrutacyjnych	93
Pytania o motywację	93
Pytania o zmiany pracy	96
Pytania o słabe i mocne strony	98
Pytania o plany zawodowe	100
Pytania o finanse	101
Pytania i praktyki na granicy prawa i manipulacji	103
Głupie pytania rekrutacyjne	106
Część II	109
Rozdział 8. Programowanie obiektowe	111
Rozdział 9. C#	123
Co każdy wiedzieć powinien?	123
Co wymaga trochę praktyki?	145
Czym można się pochwalić?	156

Rozdział 10. ASP.NET MVC	159
Co każdy wiedzieć powinien?	159
Co wymaga trochę praktyki?	173
Rozdział 11. ASP.NET Web API	181
Co każdy wiedzieć powinien?	181
Co wymaga trochę praktyki?	190
Rozdział 12. Windows Forms	193
Co każdy wiedzieć powinien?	193
Co wymaga trochę praktyki?	200
Czym można się pochwalić?	205
Rozdział 13. Windows Presentation Foundation	207
Co każdy wiedzieć powinien?	207
Co wymaga trochę praktyki?	219
Rozdział 14. Architektura aplikacji	225
Rozdział 15. Bazy danych i SQL	243
Co każdy wiedzieć powinien?	243
Co wymaga trochę praktyki?	255
Rozdział 16. HTML/CSS	263
Co każdy wiedzieć powinien?	263
Co wymaga trochę praktyki?	267
Czym można się pochwalić?	270
Rozdział 17. JavaScript	273
Co każdy wiedzieć powinien?	273
Co wymaga trochę praktyki?	280
Czym można się pochwalić?	284
Rozdział 18. EcmaScript 2015+	289
Co każdy wiedzieć powinien?	289
Co wymaga trochę praktyki?	294
Czym można się pochwalić?	297
Rozdział 19. TypeScript	301
Co każdy wiedzieć powinien?	301
Co wymaga trochę praktyki?	305
Czym można się pochwalić?	310

Rozdział 20. Angular	315
Co każdy wiedzieć powinien?	315
Co wymaga trochę praktyki?	318
Czym można się pochwalić?	322
Rozdział 21. React	327
Co każdy wiedzieć powinien?	327
Co wymaga trochę praktyki?	332
Czym można się pochwalić?	336
Rozdział 22. Infrastruktura frontendu	339
Co każdy wiedzieć powinien?	339
Co wymaga trochę praktyki?	342
Skorowidz	345

Rozdział 4.

Jak współpracować z rekruterami

Może być Twoim największym sprzymierzeńcem w drodze do wymarzonego stanowiska, może też być główną blokadą, czasem wręcz na dobre uniemożliwiającą dostanie się do danej firmy. Choć zdecydowanie częściej będzie Ci kibicował i jako programiście (a więc przedstawicielowi zawodu deficytowego na rynku pracy) wybaczy wiele, pewnych rzeczy nie zaakceptuje. Specjalista ds. rekrutacji, konsultant ds. rekrutacji, headhunter — po prostu rekruter. Jeśli zrozumiesz jego perspektywę i nauczysz się z nim współpracować, w rekrutacjach będzie Ci szło łatwiej. W tym rozdziale napiszę, co i jak powinieneś robić, by to osiągnąć.

Rodzaje rekruterów

Najbardziej podstawowy, najbardziej przejrzysty i najbardziej istotny dla Twoich rekrutacyjnych losów podział to podział na rekruterów zewnętrznych i wewnętrznych. Ci pierwsi, częściej mający w nazwie swojego stanowiska słowo „konsultant”, nazywani są zewnętrznymi dlatego, że nie rekrutują dla firmy, w której pracują, ale dla innych firm, dla swoich klientów. Ci drudzy rekrutują „wewnętrznie”, tzn. na potrzeby swej własnej organizacji. Są pracownikami wewnętrznych działów HR, nazwy ich stanowisk brzmią najczęściej „Recruitment Specialist” (czy specjalista ds. rekrutacji) albo „Sourcing Specialist”, ewentualnie „In-House Recruiter”.

Rekruterzy zewnętrzni rekrutują zwykle dla więcej niż jednej firmy jednocześnie. Wynagradzani są w głównej mierze za to, że ich klient zatrudni wybranego przez nich kandydata. Im więcej kandydatów zostanie zatrudnionych, tym więcej zarobią. Niestety ma to czasem negatywne konsekwencje: podchodząc ilościowo (czy targetowo) do swoich rekrutacji, zapominają o jakości — i to zarówno w odniesieniu do klientów (firm), jak i kandydatów, a więc potencjalnie i do Ciebie. Jeśli chodzi o firmy, bywa, że zasypują je niewłaściwymi, niezmotywowanymi lub wręcz niekompetentnymi kandydatami, w skrajnych przypadkach takimi, których w ogóle nie

widzieli na oczy. Niska jakość pracy w stosunku do kandydata najczęściej z kolei objawia się tym, że nie jest on informowany o stanowisku, na jakie jego kandydatura została zgłoszona; niewiele wie o statusie swojej rekrutacji w firmie docelowej, a czasem w ogóle nie dowiaduje się o wyniku rekrutacji.

Dużym atutem rekruterów zewnętrznych w stosunku do wewnętrznych jest to, że — skoro pracują dla kilku firm jednocześnie — dysponują większą liczbą dostępnych stanowisk, które mogą Ci zaproponować. Nawiązanie dobrej relacji z takim rekruterem, jeśli oczywiście odpowiada Ci jakość jego pracy, może w przyszłości przyczynić się do tego, że będziesz mieć dostęp do większej puli ciekawych stanowisk, czasem niedostępnych dla ogółu kandydatów (bo np. nieogłoszanych w serwisach rekrutacyjnych).

Rekruterzy wewnętrzni zwykle lepiej znają stanowisko, na które poszukują osoby. Ma to ten niebagatelny atut, że od początku są w stanie przekazać Ci więcej informacji o nim, potencjalnie oszczędzając Ci czasu, bo jeśli będziesz znał szczegóły, nie zaangażujesz się w rekrutację na stanowisko, które ostatecznie okazałoby się dla Ciebie mało ciekawe. Sami są też w stanie lepiej ocenić adekwatność Twojej kandydatury w kontekście danego stanowiska. Jeśli więc intensywnie promują Cię w swojej rekrutacji, oznacza to prawdopodobnie, że Twoje szanse w niej są większe niż w sytuacji, gdyby promował Cię rekruter zewnętrzny, gdyż ten zazwyczaj wie mniej o kulisach całej rekrutacji, np. o tym, jakich osobowościowo kandydatów woli osoba ostatecznie decydująca o zatrudnieniu. Rekruterzy wewnętrzni z kolei mają zwykle mniej do zaoferowania, jeśli chodzi o liczbę otwartych stanowisk, na które rekrutują. Oni również nie są w większości firm wolni od targetów i liczbowych wskaźników efektywności (tzw. KPI), co oznacza, że i rekruter wewnętrzny — jeśli ma niskie standardy pracy — może pozostawić Cię bez informacji zwrotnej po zakończonej rekrutacji.

Podział na rekruterów zewnętrznych i wewnętrznych jest dość ogólny i umowny. Bywają przypadki, w których trudno jednoznacznie danego rekrutera zakwalifikować do jednej z tych grup. Nie można też stwierdzić, czy lepiej jest mieć do czynienia z rekruterem pierwszego, czy drugiego rodzaju. Czasem lepiej z tym, czasem z tym. To, co liczy się przede wszystkim, to by mieć styczność z dobrym rekruterem.

Jak rozpoznać dobrego rekrutera?

Dobry rekruter może być dla Ciebie wsparciem w znalezieniu dobrej pracy i w rozwoju zawodowym. Zły rekruter może co najmniej zmarnować Twój czas, w gorszych przypadkach narazić Cię na nieprzyjemności, a w skrajnych — poważnie popsuć Twoją sytuację zawodową. Widziałem już przypadki rekruterów przesyłających swoim klientom (zatrudniającym menedżerom) przez nieuwagę CV ich podwładnych, którzy wcześniej do tych rekruterów zaaplikowali. Znam przypadek rekruterki, która

próbowała namówić do zaaplikowania na pewne stanowisko kandydata, na którego miejsce rekrutacja się toczyła (miał być zwolniony, o czym jeszcze nie wiedział). Nieraz słyszałem o rekruterach próbujących manipulować oczekiwaniami finansowymi swoich kandydatów, by w ten sposób zwiększyć swoją prowizję za udane zatrudnienie. I tak dalej.

Łatwo jest zostać rekruterem. Być dobrym rekruterem — zdecydowanie trudniej. Jest wiele cech definiujących dobrego rekrutera. Pojęcie to zmienia się też w czasie. Obserwuję na przykład, że coraz częściej firmy z branży rekrutacyjnej definiują dobrego rekrutera jako dobrego sprzedawcę, potrafiącego sprzedać firmie kandydata, a kandydatowi — daną ofertę pracy. Osobiście daleko mi do tego podejścia. Dla mnie dobry rekruter to taki, który — w dużym skrócie — jest w stanie względnie szybko znaleźć odpowiedniego kandydata, potem odpowiednio zweryfikować jego kompetencje, następnie zbudować z nim bazującą na zaufaniu i szczerości relację, by w efekcie, jeśli wszystkie pozostałe czynniki będą temu sprzyjać, doprowadzić do zatrudnienia tego kandydata skutkującego zadowoleniem jego oraz firmy, która go zatrudniła.

Dobry rekruter, po pierwsze i najważniejsze, zna branżę, w której rekrutuje. Dlatego dobry rekruter IT powinien:

- Wiedzieć, jakie są główne współcześnie wykorzystywane technologie (zwłaszcza języki programowania, frameworki, narzędzia developerskie) i jakie powiązania istnieją między nimi.
- Znać główne role, jakie odgrywają osoby zaangażowane w projekt informatyczny (programista, tester, architekt, analityk, project manager, Scrum Master).
- Znać terminologię związaną z projektem IT oraz organizacją pracy programistów (np. Scrum, code review, środowisko produkcyjne itd.).

Po co to rekruterowi i po co Tobie rekruter, który to wie? Powód jest prosty. Bez tej wiedzy rekruter nawet po przeczytaniu Twojego CV czy profilu na LinkedIn nie będzie wiedział, czy jesteś dobrym kandydatem na stanowisko, na które rekrutuje. Konsekwencją braku podstawowej wiedzy IT wśród wielu przedwcześnie biorących się za rekrutację rekruterów jest to, że specjaliści IT otrzymują mnóstwo niepasujących do ich profili zawodowych ofert. Dobrze, jeśli sytuacja wyjaśnia się na wczesnych etapach rekrutacji, a nie dopiero na spotkaniu rekrutacyjnym, na które kandydat poświęcił swój czas.

Wiedza na temat podstawowych pojęć pojawiających się w CV pozwala rekruterowi być rekruterem przyzwoitym. Rekruter naprawdę dobry będzie w stanie dodatkowo:

- Rozpoznać, czy programista stosuje dobre praktyki developerskie albo przynajmniej jest ich świadomy.

- Stwierdzić, czy programista jest na bieżąco z nowymi technologiami i czy jest to aspekt ważny w jego karierze.
- Określić, co — poza finansami i innymi oczywistymi czynnikami — decyduje o satysfakcji zawodowej programisty.

Po co rekruter ma to wiedzieć? Po to, by prowadzić z Tobą względnie merytoryczną rozmowę rekrutacyjną, na podstawie której będzie w stanie ocenić, na ile możesz być — jeszcze nie pod względem umiejętności czysto technicznych, ale już pod względem dopasowania do tego, jak programuje i jak pracuje się w danym miejscu — odpowiednim kandydatem w danej rekrutacji. Jeśli np. duże znaczenie ma dla Ciebie praca z nowymi technologiami i w wysokich standardach developerskich (np. TDD, code review czy continuous integration), a firma, dla której w danym momencie rekruter poszukuje osoby, nie ma tego w swoich projektach, taki rekruter będzie wiedział, że nie jesteś tu najlepszym kandydatem i jest ryzyko, że odejdziesz z takiego miejsca po krótkim czasie. Dobry, mający odpowiednie doświadczenie i wiedzę rekruter będzie w stanie podczas rozmowy z Tobą takie rzeczy wywnioskować. Jako osoba operująca swobodnie terminologią projektową i programistyczną szybko rozpoznasz, czy masz do czynienia z właśnie takim rekruterem.

Dalej, pracujący zgodnie z wysokimi standardami rekruter świetnie zna swojego klienta i stanowisko, na które poszukuje osoby. Nie tylko po to, by dobrze identyfikować potencjalnych kandydatów, ale też po to, by dostarczając Ci wszelkich dostępnych w danym momencie informacji, pomóc Ci zdecydować, czy oferta jest dla Ciebie — przynajmniej wstępnie — atrakcyjna. Jeśli tak i jeśli rozpocząłeś udział w rekrutacji, dobra wiedza rekrutera o jego kliencie może pomóc Ci w bardziej skutecznym zaprezentowaniu się w decydujących etapach rekrutacji. Nie chodzi tu o to, że rekruter będzie nieuczciwy wobec swojego klienta i poinformuje Cię np., jakie pytania będą zadawane. To, poza tym, że byłoby nieuczciwe, byłoby też krótkowzroczne. Ale już wskazanie Ci, byś przed rozmową techniczną odświeżył np. wiedzę akademicką z algorytmiki, może być zupełnie dopuszczalną i skuteczną pomocą w Twoich zmaganiach rekrutacyjnych. To jeden z tych momentów, które miałem na myśli, kiedy pisałem, że dobry — bo znający swojego klienta — rekruter może być Twoim sprzymierzeńcem, a przynajmniej doradcą.

Po czym jeszcze poznasz dobrego rekrutera? Czym się wyróżni?

- Wysokimi standardami w komunikacji. Napisze e-mail, który da się zrozumieć. Nie przejdzie na „ty” podczas pierwszej rozmowy (choć to podobno się zmienia). Nie poprosi Cię w pierwszym e-mailu na LinkedIn o numer telefonu, jeśli jeszcze nic konkretnego Ci nie zaoferował.
- Dotrzymywaniem słowa i terminów. Zmiana pracy to zbyt stresujące i pełne niepewności przedsięwzięcie, by zaufać komuś, kto nie dotrzymuje podstawowych obietnic (np. obiecuje, że zadzwoni danego dnia, i nie robi tego).

- Daną na czas i — o ile to możliwe — pełną, zawierającą uzasadnienie decyzji, informacją o wyniku rekrutacji. To dla mnie rekruterska świętość. Rekruter zostawiający swego kandydata bez żadnej informacji po spotkaniu rekrutacyjnym (a nawet po przesłaniu na swoją prośbę CV) zasługuje na przynajmniej czasowe bezrobocie, by samemu pozмагаć się z takimi jak on rekruterami.

Jak możesz pomóc rekruterowi?

By rekruter mógł pomóc Ci w otrzymaniu stanowiska, o które się starasz, by mógł być Twoim sojusznikiem, a nie problemem w procesie rekrutacji, Ty również powinieneś mu pomóc. Przede wszystkim bądź otwarty i szczery, jeśli chodzi o Twoje oczekiwania co do stanowiska. Mam na myśli wszelkie oczekiwania, zaczynając od finansowych, poprzez te dotyczące Twoich zadań w przyszłej pracy, a skończywszy na aspektach związanych z Twoją dalszą ścieżką kariery. Dla przykładu, jeśli w planujesz w ciągu kolejnych kilku lat objąć stanowisko team leadera zespołu programistów, ale nie jesteś pewien, czy aktualnie oferowane Ci stanowisko stwarza szanse takiego awansu w przyszłości, dobrym pomysłem byłoby podzielenie się tymi wątpliwościami z rekruterem. Pomogasz mu w ten sposób, gdyż wiedząc o Twoich aspiracjach, będzie on mógł lepiej dopasować dla Ciebie swoje przyszłe oferty.

Owsem, istnieje niebezpieczeństwo, że rekruter — powodowany chęcią zysku (w przypadku rekruterów zewnętrznych) lub chęcią ukończenia trudnej rekrutacji (w przypadku rekruterów wewnętrznych) — może chcieć podkoloryzować rzeczywistość i zaprezentować oferowane stanowisko jako lepsze, niż jest ono w istocie, ale tylko do pewnego stopnia. Rekruterzy jak ognia boją się sytuacji, w których zrekrutowany przez nich kandydat rezygnuje z pracy zaraz po jej rozpoczęciu. Oznacza to, że być może — być może, a nie na pewno, gdyż mamy do czynienia z ludźmi, czyli materią nie do końca przewidywalną — źle wykonali swoją pracę, bo niewystarczająco sprawdzili motywację i oczekiwania kandydata. Rekruter zewnętrzny po kilku takich sytuacjach traci klienta, rekruter wewnętrzny — zaufanie swoich klientów wewnętrznych, a może i pracę.

By pomóc rekruterowi, powinieneś też być wobec niego szczery, jeśli chodzi o ewentualne inne rekrutacje, w które jesteś zaangażowany w tym samym czasie, gdy uczestniczysz w prowadzonej przez niego rekrutacji. Choćby po to, by pomóc mu tak skoordynować jego rekrutację, aby zakończyła się w momencie pozwalającym Ci — w przypadku pozytywnego wyniku tej rekrutacji i pozostałych — na podjęcie decyzji, którą ofertę finalnie wybierzesz. Jeśli dodatkowo będziesz z nim szczery w kwestii finansów oferowanych Ci w ramach innych rekrutacji, może to pomóc rekruterowi w wynegocjowaniu — niejako w Twoim imieniu — wyższej stawki dla Ciebie u jego klienta. Pozostając przy finansach, na pewno wskazana będzie z Twojej

strony stałość, jeśli chodzi o Twoje oczekiwania finansowe w ciągu całego procesu rekrutacji. Zwiększanie oczekiwań w ostatniej chwili zaszkodzi — i rekruterowi, i Tobie.

Pomożesz też na pewno terminowością i generalną przewidywalnością. Permanentne odwoływanie spotkań w ostatniej chwili, niewywiązywanie się ze zobowiązań (nawet drobnych), co do których dałeś słowo, trudny kontakt i nieodpowiadanie na e-maile czy telefony zdecydowanie nie ułatwią pracy rekruterowi i nie pomogą w doprowadzeniu do pozytywnego sfinalizowania rekrutacji, w której uczestniczysz.

Jak rekruter może pomóc Tobie?

Coś za coś. Jeśli dałeś rekruterowi sporo od siebie, masz prawo oczekiwać rewanżu i jego wydatnej pomocy w Twoim ubieganiu się o pracę. Nie tylko o rewanż oczywiście chodzi, bo kompetencji i profesjonalizmu (opisanych w podrozdziale „Jak rozpoznać dobrego rekrutera?”) możesz wymagać niezależnie od tego, co dałeś rekruterowi. W relacjach między ludźmi działa jednak psychologiczna reguła wzajemności: osoba, która coś od nas otrzymała, będzie bardziej skłonna do zrewanżowania się w ten czy inny sposób.

Rekruter może przyczynić do większego, niż oczekiwałeś, podniesienia poziomu Twoich zarobków. Jeśli w rozmowie o swoich oczekiwaniach finansowych podałeś kwotę znacząco niższą niż przewidywany budżet na danym stanowisku — bo np. po wielu latach pracy w jednej firmie i w związku z niechodzeniem na rozmowy rekrutacyjne straciłeś kontakt z rynkiem i nie orientujesz się, ile możesz za swoje kompetencje oczekiwać — rekruter może zaproponować Ci poziom wynagrodzenia wyższy, niż oczekiwałeś. Czy będzie działał wtedy na szkodę swojego klienta? Nie sądzę. Jeśli bowiem nie zasugeruje Ci podwyższenia oczekiwań i doprowadzi do zaoferowania Ci wynagrodzenia takiego, jakiego pierwotnie oczekiwałeś — czyli zaniżonego w stosunku do płac rynkowych — ryzykuje, że szybko odejdziesz po zreflektowaniu się, ile rzeczywiście możesz na rynku otrzymać.

Jak już wspomniałem, rekruter — jeśli już rozpozna w Tobie kandydata, którego warto promować w dalszych etapach rekrutacji — może być bezcennym wsparciem i wskazać Ci, co zrobić, by wypaść w tych kolejnych etapach jak najlepiej. Widział prawdopodobnie nie raz rozmowy rekrutacyjne prowadzone w danej firmie; wie, kim są ludzie decydujący o zatrudnieniu; zna błędy, jakie popełniali dotychczasowi kandydaci; wie o Twoich błędach z pierwszego etapu i zdaje sobie sprawę, czy i na ile są one do wyeliminowania w etapach decydujących.

Jeśli bieżąca rekrutacja z jakichś względów — mogą ich być w rekrutacjach tysiące i nie zawsze o czysto merytorycznym charakterze — nie zakończyła się pomyślnie, w przyszłości rekruter może zaproponować Ci udział w rekrutacji na inne ciekawe

stanowiska. Jak już wcześniej wspominałem, może mieć dostęp do ofert, które nie są dostępne powszechnie (np. w serwisach rekrutacyjnych), dzięki czemu są cenniejsze.

Dobry, doświadczony w danym sektorze rekruter to także bezcenne źródło informacji i wiedzy na temat tego sektora właśnie. Nawiązując relację z takim rekruterem, możesz liczyć np. na informację o tym, jakie są stawki wynagrodzenia na rynku w Twojej specjalizacji; jaki framework zdobywa popularność, sądząc po zapotrzebowaniu jego klientów na osoby o konkretnych kompetencjach; w której firmie nie warto się zatrudniać ze względu na stare technologie i niską jakość praktyk programistycznych itd. Może być więc dla Ciebie nie tylko wsparciem w danym procesie rekrutacyjnym, może też być tymczasowym — ale o potencjalnie długofalowych skutkach — doradcą zawodowym, podpowiadającym możliwe kierunki rozwoju i metody ich realizowania.

Za co można trafić na czarną listę u rekrutera?

Jako programista reprezentujesz pożądaną na rynku zawód, którego przedstawiciele są rozchwytywani przez pracodawców i cenieni przez rekruterów. Zaryzykuję tezę, że wybaczone Ci będzie jako kandydatowi więcej niż kandydatom z mniej wziętych profesji. Ale i programista ma — a przynajmniej powinien mieć — wyznaczone granice tolerowanego zachowania podczas rekrutacji. Po przekroczeniu tych granic, czyli po popełnieniu niewybaczalnych z punktu widzenia rekrutera grzechów, możesz znaleźć się w danej firmie lub u danego rekrutera na tzw. czarnej liście. Nie we wszystkich firmach coś takiego funkcjonuje, firmy lepiej lub gorzej radzą sobie z dokumentowaniem procesów rekrutacyjnych i uczestniczących w nich kandydatów. W każdym razie, jeśli gdzieś taka lista by się znalazła, wskazuję tu najbardziej typowe powody, które mogłyby spowodować, że zostaniesz na niej umieszczony.

- Skrajnie nieuprzejme, aroganckie, a nawet chamskie zaprezentowanie się na rozmowie rekrutacyjnej. Chyba nie wymaga bardziej obszernego komentarza. Jeśli nieodpowiednio zachowałeś się akurat w czasie, w którym powinieneś był zachować się szczególnie odpowiednio, mało kto da Ci drugą szansę, nawet jeśli jesteś geniuszem programistycznym.
- Rozpoczęcie rekrutacji za pośrednictwem firmy rekrutacyjnej, a później bezpośrednio zaaplikowanie do firmy docelowej — klienta tej firmy rekrutacyjnej — w tajemnicy przed firmą rekrutacyjną. Miałoby to na celu uniknięcie przez klienta zapłacenia firmie rekrutacyjnej wynagrodzenia za zrekrutowanie Ciebie (bo w końcu sam do klienta zaaplikowałeś) albo przyspieszenie rekrutacji (bo w sumie bezpośrednio zaaplikowałeś). Niezależnie od motywacji, to zachowanie jest bardzo nie w porządku.

- Podanie na finalnej rozmowie rekrutacyjnej u klienta oczekiwań finansowych wyższych niż te, które podało się we wstępnej rozmowie z rekruterem. Być może — gdy już witasz się z gąską i czujesz, że blisko jest do pozytywnej decyzji ze strony firmy — kuszące jest dodanie sobie jakiejś kwoty do pierwotnie określonych oczekiwań. Grozi to jednak odebraniem Cię jako kandydata niewiarygodnego. Rekrutacja to czas obustronnego sprawdzania się. Jeśli przeskrobałeś coś w jej trakcie, druga strona ma prawo przypuszczać, że będzie się to z Twojej strony powtarzać.
- Niezjawienie się na rozmowie rekrutacyjnej — bez uprzedzenia, bez podania uzasadnienia nieobecności, z późniejszym nieodbieraniem telefonu jako sposobem na uniknięcie „napastującego” rekrutera. Nie zdziw się, jeśli później już nigdy nie zadzwoni.
- Wyraźne i bezdyskusyjne kłamstwo w CV dotyczące istotnych zawartych w nim faktów, np. dat w historii zatrudnienia. Rozmawiałem kiedyś z kandydatem, który spytany przeze mnie o pewne nieścisłości w datach w jego CV, otwarcie przyznał, że skłamał, jeśli chodzi o czas swojego uczestnictwa w projektach, bo sądził, że „tak będzie korzystniej wyglądało”. Powtórzę: rekrutacja to krótki czas obustronnego sprawdzania się. Jeśli po dwudziestu minutach znajomości wychodzi na jaw, że kłamiesz w oficjalnym dokumencie, znajomość nie ma szansy trwać wiele dłużej.

Moi najbardziej pamiętni kandydaci

Kończąc rozdział o możliwościach współpracy z rekruterami, chciałbym wspomnieć swoich dawnych kandydatów, których zdecydowanie za daleko posunięty ekscentryzm uniemożliwił mi jakkolwiek z nimi współpracę. Choć spotkałem ich lata temu, w mej pamięci pozostali do dziś.

Pan Michał, biznesmen i założyciel start-upów, który przyniósł na rozmowę rekrutacyjną swojego laptopa, uruchomił go i przez całą rozmowę pracował na nim, coś czytał, coś pisał, a w międzyczasie — jakby przy okazji! — odpowiadał na moje pytania. Byłem jeszcze wówczas początkującym rekruterem i pewnie dlatego nie zdobyłem się na to, by zakończyć rozmowę po dziesięciu minutach. Gdy odwiedzam dziś profil pana Michała na LinkedIn, widzę, że jego błyskotliwej kariery (liczne stanowiska typu Director, President itp.) nie zatrzymało moje ówczesne odrzucenie jego kandydatury.

Pan Antoni, programista, który rozmowę ze mną rozpoczął od słów: „Wreszcie pan jest. No i jakie trudne pytania będzie mi pan teraz zadawał?”. A gdy już zacząłem zadawać te swoje pytania, autentycznie w 80% przypadków nie wiedziałem, czy odpowiadając, mówi poważnie, czy żartuje. Pod koniec rozmowy odebrał telefon. Zmartwiony rozmawiał kilka minut, a po skończeniu zwrócił się do mnie: „Przepraszam, to moja dziewczyna, miała wypadek. Mam nadzieję, że nie urwało jej nóg, bo musiałbym ją opuścić. Żartuję”.

Pan Leszek, project manager, który dowiedziawszy się o mojej decyzji o niekontynuowaniu z nim rozmów, napisał mi w e-mailu: „Jestem więcej niż szczęśliwy, że nie będę musiał pracować dla bandy darmozjadów takich jak Pan i Pana firma. Cała ta branża to banda oszustów i spekulantów (...). Napisałem też maila do szefa Pana firmy, sugerując, aby posłał Pana na kursy wymowy angielskiej, bo to co Pan reprezentuje jest żenujące i dyskwalifikuje Pana w roli interviewera”.

Pan Piotr, programista, który zaaplikował do mojej firmy dwa razy w ciągu kilku miesięcy, przy czym za drugim razem miał zupełnie inne nazwisko niż za pierwszym (wszystkie pozostałe rzeczy — imię, e-mail, telefon, doświadczenie — były takie same). Za pierwszym razem odpadł ze względu na skrajnie aroganckie zachowanie; drugiej szansy, choć występował pod innym nazwiskiem, już nie otrzymał.

Pan Tomasz, administrator Linuksa, który na rozmowie dał się zapamiętać głównie tym, że miał kolor włosów sugerujący jakieś daleko posunięte i nie do końca udane eksperymenty z ogniem. Ale tak naprawdę wstawił się zachowaniem po spotkaniu, kiedy próbował odprowadzić do domu — bez jej wiedzy, a tym bardziej zgody — moją koleżankę z zespołu rekrutacji, która również uczestniczyła w spotkaniu.

Skorowidz

A

- abstract factory, *Patrz:* fabryka abstrakcyjna
- abstrakcja, 112
- adapter, 235
- adres URL, 163
 - mapowanie, 159
 - przekierowanie, 164
- Agile, *Patrz:* programowanie zwinne
- AJAX, 164
 - odwołanie międzys domenowe, 173
- AJAX Helper, 170
- akcja, 334
 - kontrolera, *Patrz:* kontroler akcja
 - REST, 181
 - wynik, 172, 173
- anchoring, *Patrz:* zakotwiczenie
- Angular, 315
- Angular CLI, 323
- AngularJS, 315
- aplikacja
 - CRUD, *Patrz:* CRUD
 - graficzna, 193
 - typu desktop, 207
 - logika, 159
 - rozproszona, 340
 - WPF, 209
 - XBAP, 223
- architektura
 - oparta na serwisach, *Patrz:* SOA
 - wielowarstwowa, 229
- AsyncPipe, 320, 321
- atak
 - Cross Site Scripting, *Patrz:* atak XSS
 - CSRF, 343
 - SQL Injection, 254
 - XSS, 343

- atrybut, 130
 - ActionName, 166, 183
 - AllowHtml, 176
 - AttributeUsage, 131
 - Bind, 167
 - ChildActionOnly, 179
 - class, 329
 - className, 329
 - DataType, 167
 - DisplayFormat, 167
 - DisplayName, 167
 - Flags, 145
 - FromBody, 189
 - FromUri, 189, 192
 - HttpDelete, 166
 - HttpGet, 166
 - HttpPost, 166
 - HttpPut, 166
 - InternalsVisibleTo, 148
 - MaxLength, 167
 - name, 220
 - onClick, 331
 - Range, 167
 - RegularExpression, 167
 - Required, 167
 - ScaffoldColumn, 167
 - StringLength, 167
 - tworzenie, 131
 - ValidateInput, 176
 - x:name, 220
 - XmlArrayItem, 156
 - xmlns, 219
- Autofac, 230
- autoryzacja, 174

B

Babel, 340
 backend, 47
 banana in the box, 318
 baza danych
 atak, 254
 master, 257
 model, 257
 msdb, 257
 normalizacja, 260
 pole, 243
 postać normalna, 260
 procedura składowana, 249
 rekord, 243
 relacja, 244
 schemat, 250
 statystyka, 256
 systemowa, 257
 tabela, 244
 tempdb, 257
 transakcja, 244, 245, 252
 izolowanie, 254
 widok, 248
 zarządzanie, *Patrz:* DBMS
 BEM, 271
 biblioteka
 DirectX, 207
 Entity Framework, 231
 Hibernate, 231
 jQuery, 283
 Microsoft.Web.Optimization.dll, 168
 nHibernate, 231
 NPM, 341
 ORM, 231
 React, 337, *Patrz:* React
 Redux, 334
 RxJS, 319, 325
 box model, 263
 boxing, 147
 bridge, *Patrz:* most
 BSON, 190
 budowniczy, 235
 buforowanie, 176
 builder, *Patrz:* budowniczy
 bundle, 168

C

caching, *Patrz:* buforowanie
 callback, 283, 284, 290, 291, 300, 320, 331
 zagnieżdżanie, 283, 296
 callback hell, 283, 296
 Castle Windsor, 230
 CDN, 173
 chain of responsibility, *Patrz:* łańcuch
 zobowiązań
 change detection, *Patrz:* mechanizm
 wykrywania zmian
 CIL, 123
 closure, *Patrz:* domknięcie
 code splitting, 342
 cohesion, *Patrz:* oprogramowanie spójność
 słowo kluczowe, 254
 command, *Patrz:* polecenie
 Command Query Responsibility Segregation,
 Patrz: wzorzec CQRS
 composite, *Patrz:* kompozyt
 conditional type, *Patrz:* typ warunkowy
 constraint, *Patrz:* więzy spójności
 Content Delivery Network, *Patrz:* CDN
 content security policy, *Patrz:* CSP
 corridor test, *Patrz:* test korytarzowy
 CORS, 173
 Crockodilos Nick, 75
 CRUD, 174, 227
 CSP, 343
 CSS, 263
 BEM, 271
 hierarchia ważności reguł, 264
 preprocesor, *Patrz:* preprocesor CSS
 CSS Grid, 267, 268
 CSS specificity, *Patrz:* CSS hierarchia
 ważności reguł
 CV, 51, 52, 78
 cel zawodowy, 58
 certyfikaty, 55
 dane kontaktowe, 55, 57
 długość, 56
 doświadczenie, 53, 55, 56, 57
 niewielkie, 60
 elementy zbędne, 58
 kompetencje, 55, 57
 kompletność, 53, 54, 56
 komunikatywność, 54, 55

- motto, 58
- oczekiwania finansowe, 58
- otwieralność, 57
- profil, 55, 58
- przejrzystość, 55, 56
- referencje, 58, 89, 105
- repozytorium kodu, 58
- wykształcenie, 55
- zdjęcie, 57
- zwięzłość, 56

D

- danych wiązanie, *Patrz:* data binding
- data binding, 316, 318, 321
 - szablon, 317
- database lock, 245
- DBMS, 243
- deadlock, *Patrz:* zakleszczenie
- debugger, 144
- deklaracja DOCTYPE, 270
- dekorator, 235, 238
 - @Injectable, 317
 - @Input, 318
 - @Pipe, 324
 - @ViewChild, 321
- delegat, 119, 146
 - anonimowy, 149
 - multiemisji, 146
 - ogólny, 127
 - ThreadStart, 153
- dependency injection, *Patrz:* wstrzykiwanie zależności
- deserializacja, 121
- deskryptor, 287
- destruktor, 116, 117
- DI, *Patrz:* wstrzykiwanie zależności
- discriminated union type, 311
- docking, *Patrz:* dokowanie
- DOCTYPE, 270
- document flow, *Patrz:* przepływ dokumentu
- dokowanie, 194, 210, 213
- DOM, 316, 317, 319
- Domain Driven Design, *Patrz:* technika DDD
- domknięcie, 276
- drzewo, 221
- dyrektywa, 317, 322
 - atrybutowa, 317
 - class, 318

- ngClass, 318
- ngFor, 317, 319
- ngIf, 317
- ngModel, 318
- preprocesora, 144
- strukturalna, 317
- using, 124
- dyskryminacja, 103
- dziedziczenie, 112, 114, 132, 136
 - hierarchiczne, 118
 - pojedyncze, 118
 - prototypowe, 280
 - wielobazowe, 118
 - wielopoziomowe, 118

E

- EcmaScript, 289
- element
 - ToolTip, 223
 - Trigger, 222
 - x:Code, 224
- em, 203
- enkapsulacja, *Patrz:* hermetyzacja
- Enterprise Service Bus, *Patrz:* ESB
- enumeracja, 305, 306
- ES6, *Patrz:* standard ES2015
- ESB, 230
- etat, *Patrz:* umowa o pracę
- event binding, 318, 321
- event loop, *Patrz:* pętla zdarzeń
- exclusive lock, 245

F

- fabryka abstrakcyjna, 235, 236
- factory method, *Patrz:* metoda wytwórcza
- falsy, 278
- fasada, 235, 239
- filtr, 174
 - Action Filter, 174
 - akcji, 188
 - Authentication Filter, 174
 - Authorization Filter, 174
 - Exception Filters, 175
 - kolejność, 175
 - obsługi wyjątków, 184, 185
 - rejestrowanie, 186, 187
 - OutputCache, 177

filtr
 rejestrowanie, 175
 Result Filter, 175
 uwierzytelnienia, 191

firma
 czas istnienia, 45
 dress code, 82
 informatyczna, 44, 45
 kontraktorska, 49
 kultura organizacyjna, 42
 nieinformatyczna, 44, 45
 polska, 42
 profil, 44, 45
 rotacja pracowników, 76, 77, 87
 technologiczna, 44
 wielkość, 43
 zagraniczna, 42, 46
 zdobywanie informacji, 76

Flexbox, 267, 268

forma zatrudnienia, 48

format
 BAML, 223
 BSON, 190
 JSON, 169, 181, 182, 190
 XAML, 207, 211, 214, 221
 XML, 181, 182

formularz, 194, 200, 334
 MDI, 195
 reaktywny, 325
 szablonowy, 325

Fowler Martin, 227

Franklin Benjamin, 73

frontend, 47, 339

full-stack developer, 47

funkcja
 add, 285
 Array.map, 290
 AVG, 253
 COALESCE, 258
 COUNT, 253
 czysta, 330
 DELETE, 258
 DROP, 259
 eval, 279
 grupująca, 253
 HOC, 337
 ISNULL, 258
 MAX, 253
 MIN, 253

NEWID, 249
 Object.create, 280
 Object.defineProperty, 287
 RenderBody, 171
 RenderPage, 171
 ROW_NUMBER, 257
 rozwijanie, 285
 SUM, 253
 super, 295
 TRUNCATE, 259
 tworzenie, 291
 wykonująca się natychmiast, 275
 zagnieżdżona, 276
 zasięg, 275

G

generator, 298
 generic class, *Patrz:* klasa generyczna
 GitHub, 78
 GoldenLine, 32
 grid layout, 267
 Guice, 230
 GUID, 142, 249

H

HackerRank, 51, 52
 hermetyzacja, 112, 116
 Higher Order Component, *Patrz:*
 komponent wyższego rzędu
 hoisting, *Patrz:* windowanie
 hook useState, 335
 HTML, 263
 encja, 269
 semantyczny, 268
 HyperText Markup Language, *Patrz:* HTML

I

Immediately Invoked Function Expression,
Patrz: IIFE
 impure pipe, 324
 indeksy, 141
 instrukcja
 break, 129
 catch, 140
 continue, 129
 export, 294

- import, 294
- lock, 138
- return, 139
- throw, 139
- throw exception, 139
- try .. catch, 139
- using, 124
- yield break, 148
- yield return, 148
- interfejs, 120, 121, 159, 304, 305
 - CRUD, 181
 - dostępność, 149
 - IDispisable, 152
 - IDisposable, 124
 - IModelBinder, 177
 - implementacja, 132, 134
 - IMultiValueConverter, 221
 - INotifyPropertyChanged, 216
 - kompilacja, 305
 - użytkownika, 159, 337
 - React, 327
- interpreter, 236
- intersection type, 310
- inversion of control, *Patrz:* odwrócenie zależności
- IoC, *Patrz:* odwrócenie zależności
- iterator, 236, 295
 - implementacja, 298

J

- jagged array, *Patrz:* tablica nieregularna
- JavaScript, 273, 280, 289
- język
 - C#, 123
 - CIL, *Patrz:* CIL
 - CoffeeScript, 301
 - Common Intermediate Language, *Patrz:* CIL
 - dynamicznie typowany, 275, 301
 - Flow, 301
 - interpretowany, 341
 - JavaScript, *Patrz:* JavaScript
 - jednowątkowy, 283, 284
 - programowania obiektowy, 123
 - SQL, *Patrz:* SQL
 - statycznym systemie typów, 301, 302
 - TypeScript, *Patrz:* TypeScript
 - typowanie, 301
 - zapytań strukturalny, *Patrz:* SQL

- JSONP, 173
- JSX, 329

K

- kariera, 41
- klasa, 111, 305
 - abstrakcyjna, 120, 121, 304
 - Array, 124
 - ArrayList, 124, 156
 - bazowa, 114, 132
 - Canvas, 208
 - CSS, 271
 - częściowa, 129
 - ES6, 293
 - generyczna, 309
 - HtmlHelper, 163
 - HttpClient, 183
 - instancja, 111
 - Interlocked, 152
 - MenuStrip, 196
 - ModelBinders, 177
 - Monitor, 150
 - Mutex, 150, 151
 - Observable, 319
 - pochodna, 114
 - Queue, 309
 - sealed, 123
 - Semaphore, 151
 - składniki, 117
 - statyczna, 125
 - StringBuilder, 140
 - Subject, 321
 - System.Attribute, 131
 - System.ComponentModel.Sort
 - ↳ Description, 209
 - System.Configuration.AppSettingsReader, 201
 - System.Configuration.Configuration
 - ↳ Manager.AppSettings, 201
 - System.Drawing.Color, 200
 - System.Drawing.SystemIcons, 204
 - System.IO.Path, 155
 - System.Object, 135
 - System.Threading.Thread, 153
 - System.Windows.Forms.Button, 193
 - System.Windows.Forms.Control, 193, 199
 - System.Windows.Forms.Form, 200
 - TempData, 178

klasa

- Thread, 154
- ToolStripMenuItem, 196
- tworzenie, 136
- TypeScript, 304
- XMLSerializer, 156

klauzula catch, 139

klucz, 297

- główny, 245, 249
- obcy, 245
- Symbol.iterator, 297
- złożony, 245

klucz-wartość, 296

kolejka, 309

kolekcja, 117, 127, 141, 149, 295, 297

- ogólna, 145
- par klucz-wartość, 296

komenda, 215, 216

komentarz, 250

- usuwanie, 341
- znacznik, 138

kompilacja, 323

komponent, 315, 317, 322, 327

- controlled component, 334
- faza życia, 336
- funkcyjny, 328, 330
- klasowy, 328
- komunikacja, 321
- prezentacyjny, 332
- props, 328
- PureComponent, 333
- React, 327
- root, 315
- state, 328
- wyższego rzędu, 337
- zagnieżdżony, 321

kompozyt, 235

komunikat, 200

konstruktor, 115, 117, 318

- klasy bazowej, 136
- kopiujący, 116
- przeciążenie, 113, 135
- statyczny, 130

kontekst, 333

kontener, 332

- DI, 230

kontrakt, 49

kontroler, 159, 160

- akcja, 165, 169
- tworzenie, 161

kontrolka

- Adorner, 223
- Button, 193, 200
- Canvas, 209
- CheckBox, 195, 197
- CheckedListBox, 203
- ComboBox, 199, 202
- ContextMenu, 202
- do edycji tekstu, 199
- DockPanel, 210
- ErrorProvider, 204
- Grid, 209
- GroupBox, 198
- ListBox, 196, 199, 209
- listy, 199
- MaskedTextBox, 202
- MenuStrip, 196, 197
- Panel, 198
- PictureBox, 197
- RadioButton, 197
- RichTextBox, 195, 197, 201
- StackPanel, 210
- StatusStrip, 197
- TabControl, 210
- TextBox, 195, 198
 - menu kontekstowe, 203
- Timer, 198, 202
- ToolStrip, 197
- ToolStripDropDownButton, 201
- ToolStripSplitButton, 201
- tworzenie, 202
- typ
 - Content Presenter, 210
 - Control, 210
 - Panel, 210
 - Shape, 210
- UniformGrid, 209
- WindowsFormsHost, 219
- właściwości, 205, *Patrz też:* właściwość
- WrapPanel, 210

konwerter wartości, 220

krotka, 306

kursor, 259

L

layout, 164
 lazy module loading, *Patrz:* moduł ładowanie leniwe
 LESS, 268
 licencja MS-PL, 159
 liczba zmiennoprzecinkowa, 283
 Linked Server, 257
 LinkedIn, 32, 51, 52
 LINQ, 149
 lista, 196
 literal type, *Patrz:* typ literałowowy

Ł

łańcuch zobowiązań, 236, 239

M

makrozadanie, 284
 manifest Agile, 228
 mapa, 296
 mapped type, 313
 Mason Malia, 102
 mechanizm

- bezpieczeństwa
 - CSP, 343
 - same-origin policy, 342, 343
- Garbage Collection, 123, 141, 156, 299
 - sterowanie, 153
- hooks, 328, 335, *Patrz też:* hook
- mapowania adresów URL, 159
- MultiBinding, 221
- NgZone, 326
- puli wątków, *Patrz:* wątek pula uwierzytelniania, 159
- wstrzykiwania zależności, *Patrz:*
 - wstrzykiwanie zależności
 - wykrywania zmian, 323
 - zarządzania stanem aplikacji, 159

Media Formatter, 191
 mediator, 236
 menedżer pakietów, 339
 menu kreator, 199
 metadane, 130
 metoda, 112, 117

- abstrakcyjna, 120
- Add, 196
- anonimowa, 147

apply, 286
 Array.map, 330
 BeginInvoke, 146
 bind, 286, 331
 call, 285
 CheckBox, 163
 Concat, 130
 Content, 172
 ContentResult, 166

- częściowa, 129

 Debug.Assert, 142
 Delegate.Combine, 146
 Display, 163
 Dispose, 152
 DropDownList, 163
 Editor, 163
 EmptyResult, 166
 EndInvoke, 146
 Equals, 149
 Execute, 161
 FileResult, 166
 Finalize, 152
 FromArgb, 200
 get, 132
 GetFileName, 155
 GetHashCode, 149
 GetInvalidPathChars, 155
 GetValue, 201
 Hidden, 163
 HTML Helper, 163
 HttpClient.get, 320
 HttpNotFound, 166
 HttpStatusCodeResult, 166
 HttpUnauthorizedResult, 166
 InitializeComponent, 199
 Insert, 196
 Invalidate, 199
 jako pole klasy, 314
 JavaScriptResult, 166
 JsonResult, 166
 Keep, 178
 ListBox, 163
 map, 276
 MoSCoW, 227
 Move, 204
 next, 297
 ngOnDestroy, 320
 ngOnInit, 318
 Object.freeze, 286

metoda

OpenFile, 196
 otwarta na przesłonięcie, 132
 parametr, 125
 PartialViewResult, 165
 Password, 163
 Path.Combine, 155
 Peek, 178
 PerformClick, 193
 private virtual, 126
 Promise.finally, 300
 przeciążenie, 115
 przesłonięcie, 115, 132
 publiczna, 173
 QueueUserWorkItem, 154
 RadioButton, 163
 Redirect, 164
 RedirectResult, 165
 RedirectToAction, 161, 165
 RedirectToRoute, 165
 RedirectToRouteResult, 166
 Refresh, 199
 render, 328, 329, 333
 Resize, 204
 rozszerzająca, 143
 sealed, 123
 set, 132, 135
 SetItemChecked, 203
 setState, 329, 332
 shouldComponentUpdate, 333
 ShowDialog, 195
 slice, 280
 splice, 280
 statyczna, 119
 String.IsNullOrEmpty, 144
 subscribe, 320
 sygnatura, 112
 System.GC.Collect, 153
 szablonowa, 236
 TextArea, 163
 TextBox, 163
 then, 294, 296
 Thread.Abort, 154
 Thread.Interrupt, 154
 ThreadStart, 154
 ToString, 140
 ukrywanie, 133
 unsubscribe, 321
 Update, 199

ValidationSummary, 164
 View, 164
 ViewResult, 165
 void, 125
 WndProc przesłanianie, 205
 wytwórcza, 235
 middleware, 47
 mikrozadanie, 284
 minifikacja, 168, 340, 341
 mock, 232, 233
 model, 159, 160, 172, 233, 234
 Binder, 177
 hierarchia, 172
 module bundler, 340
 moduł
 core, 322
 ES6, 294
 ładowanie leniwe, 322
 shared, 322
 uwierzytelniania, 142
 modyfikator dostępu, 127
 most, 235
 motywacja, 94, 95
 Muhammad Ali Jinnah, 80
 multiemisja, 147
 Multiple Document Interface,
 Patrz: formularz MDI

N

nadklasa, 295
 napis szablonowy, 293
 null, 274, 275

O

obfuskacja, 340, 341
 obiekt, 111
 Array, 276
 CLR, 177
 docelowy, 215
 immutable, 148
 inicjalizacja, 282
 iterable, 297
 JSON, 173, 191
 Map, 296, 299
 niemutowalny, 286, 323
 Object, 296
 Observable, 297, 319

odwołanie, 119
 Promise, 294, 296, 320
 props, 328, 329
 Rectangle, 208
 rzutowalny, 128
 rzutowanie, 157
 Semaphore, 151
 Session, 165
 Set, 295, 299
 string
 łączenie, 130, 140
 TempData, 165, 178
 ViewBag, 165
 ViewData, 165
 zablokowany, 150
 źródłowy, 215
 Object-Relational Mapping, *Patrz:* ORM
 obserwator, 236
 obszar, 168
 odwiedzający, 236
 odwołanie, 119
 odwrócenie zależności, 230
 operacja arytmetyczna overflow, 155
 operator, 136
 !!, 278
 &&, 279
 ??. 128
 ||, 279
 +, 130
 ==, 277
 ===, 277
 alternatywy, 279
 APPLY, 262
 as, 128, 137
 bang bang, 278
 concatMap, 326
 double bang, 278
 is, 128
 koniunkcji, 279
 logiczny, 279
 mergeMap, 326
 new, 282, 293, 297
 przeciążenie, 114
 spread, 292
 switchMap, 326
 typeof, 277
 wildcard, 246
 oprogramowanie spójność, 227

ORM, 231
 overflow, 155
 over-posting, 192

P

pakiet menedżer, *Patrz:* menedżer pakietów
 pamięć
 podręczna, 344
 zarządzanie automatyczne, *Patrz:*
 mechanizm Garbage Collection
 parametr, 125
 rest, 291
 pełnomocnik, 235
 pętla
 for, 273
 for .. in, 274, 287, 295
 for .. of, 297, 298
 or .. of, 295
 zdarzeń, 284, 285
 Pico, 230
 PID, 121
 pipe, 321, 322, 324
 pixel, 203
 platforma
 ASP.NET MVC, 159, 160, 161, 162, 167,
 168, 184
 filtr, 174
 ruting, 169
 ASP.NET Web API, 181, 183, 184, 190
 bindowanie parametrów, 189
 błąd walidacji, 187
 filtr, 191
 ruting, 182
 ASP.NET WebForms, 181, 184, 234
 Web API, 190
 Windows Forms, 162, 193, 234
 plik
 .config, 201
 .cshtml, 164, 171
 .resX, 143
 .vbhtml, 171
 App.xaml, 211
 Assembly-Info.cs, 148
 tsconfig.json, 307
 point, 203
 pole, 117
 const, 146
 klasy, 331

- pole
 - niezmiennie, 125
 - readonly, 146
 - statyczne, 157
 - Target, 156
 - polecenie, 236
 - ng build --prod, 323
 - UPDATE_STATISTICS, 256
 - polimorfizm, 112, 115
 - polyfill, 341
 - PostCSS, 268
 - powiązanie, 215
 - pozycjonowanie, 162
 - preprocesor
 - CSS, 268
 - dyrektywa, 144
 - prezenter, 233
 - procedura składowana, 249, 261
 - proces, 121
 - Garbage Collection, *Patrz:* mechanizm Garbage Collection
 - identyfikator, *Patrz:* PID
 - Process Identifier, *Patrz:* PID
 - programowanie
 - funkcyjne, 285
 - obiektowe, 111, 112, 123, 280
 - reaktywne, 319
 - zwinne, 228
 - properta, *Patrz:* właściwość
 - property binding, 318, 321
 - protokół SOAP, *Patrz:* SOAP
 - prototyp, 235, 280
 - Array, 281
 - Function, 285
 - rozszerzanie, 281
 - proxy, *Patrz:* pełnomocnik
 - przeciążenie, 113
 - konstruktora, *Patrz:* konstruktor przeciążenie
 - metody, *Patrz:* metoda przeciążenie
 - operatora, *Patrz:* operator przeciążenie
 - przeglądarka
 - layout thrashing, 270
 - reflow, 270
 - repaint, 270
 - przepływ dokumentu, 267
 - przeźrzenie nazw, 136
 - DataAnnotations, 167
 - System.Reflection, 147
 - w pliku XAML, 219
 - przycisk submit, 179
 - przypisanie destrukuryzujące, 293
 - pseudoelement, 269
 - pseudoklasa, 269
 - pure pipe, 324
- ## R
- Razor, 163, 171
 - React element, 329
 - React Fiber, 336
 - reactive form, *Patrz:* formularz reaktywny
 - refaktoryzacja, 233
 - referencja, *Patrz też:* odwołanie
 - cykliczna, 127
 - silna, 156
 - słaba, 156
 - refleksja, 147
 - regression test, *Patrz:* test regresyjny
 - rekrutacja
 - logistyka, 79
 - opis stanowiska, 78, 79
 - programista
 - CV, *Patrz:* CV
 - wygląd, 82
 - zachowania naganne, 69, 70
 - zachowania pożądane, 67, 68, 83, 84
 - zapach, 82
 - przygotowania, 76, 77
 - rozmowa telefoniczna, 73
 - savoir-vivre, 83, 84
 - wygląd kandydata, 81
 - za pomocą aplikacji, 74
 - rekruter
 - cechy pożądane, 66, 69
 - umiejętności, 65, 66
 - wewnętrzny, 63, 64, 67
 - zewnętrzny, 63, 64, 67
 - relative binding, *Patrz:* wiązanie względne
 - renderer, 337
 - renderowanie, 336
 - responsive design, 266
 - REST, 181, 182
 - rozmowa rekrutacyjna
 - częstotliwość, 33, 38, 97
 - do kilku firm równolegle, 34
 - jako argument do negocjacji w obecnej firmie, 35, 36
 - o co nie pytać, 88, 89

- o co pytać, 87, 88
- odpowiedź, 85, 86
- praca domowa, 105
- pytanie
 - głupie, 106
 - manipulatorskie, 104, 105
 - na granicy prawa, 103
 - o finanse, 101, 102
 - o motywację, 93, 94, 95
 - o plany zawodowe, 100
 - o referencje, *Patrz:* CV referencje
 - o słabe i mocne strony, 98, 99
 - o zmiany pracy, 96, 97, 98
- w publie, 90
- ruting, 169, 182
- rzutowanie
 - bezpieczne z użyciem operatora as, 137
 - bezpośrednie, 137

S

- same-origin policy, 342, 343
- sanityzacja, 343
- Sass, 268
- scaffolding, 174
- sekcja, 170
- selektor, 264
 - #id, 264
 - .class, 264
 - attribute, 264
 - tag, 264
- serializacja, 121, 156, 192
 - binarna, 142
 - obiektu do par klucz-wartość, 190
 - właściwości obiektu, 192
 - XML, 142
- service, *Patrz:* usługa
- serwis społecznościowy, 77
- shared lock, 245
- singleton, 235, 237
- skalowalność, 230
- skoczek, 36, 37, 38
 - zarobki, 37
- słownik, 167
- słowo kluczowe
 - abstract, 120
 - ADD, 250
 - ALTER TABLE, 250, 251
 - AND, 252
 - ASC, 252
 - async, 296
 - await, 296
 - checked, 155
 - const, 125, 290, 306
 - CREATE, 251
 - DELETE, 251
 - DESC, 252
 - DISTINCT, 249
 - DROP, 251
 - dynamic, 137
 - enum, 305, 306
 - function*, 298
 - HAVING, 253
 - IDENTITY, 249
 - IN, 249
 - infer, 313
 - INSERT, 251
 - int, 134
 - jako nazwa zmiennej, 143
 - JOIN, 247
 - let, 289, 290
 - lock, 138, 150
 - new, 133
 - NOLOCK, 257
 - OR, 252
 - out, 125
 - override, 123, 132
 - params, 145
 - partial, 129
 - ref, 125
 - sealed, 123
 - SELECT, 252
 - static, 125
 - super, 295
 - this, 118, 119, 281, 285, 290, 291, 331
 - TOP, 252
 - UNION, 258
 - UNION ALL, 258
 - UPDATE, 251
 - using, 124, 150, 152
 - var, 138, 274, 289, 290
 - virtual, 132
 - void, 125, 146
 - volatile, 150
 - WHERE, 252, 253
 - yield, 148, 298
- smoke test, *Patrz:* test dymny
- SOA, 230

SOAP, 142, 182
 SOLID, 225
 sortowanie
 rodzaj, 254
 wrażliwości, 255
 source map, 342
 Spolsky Joel, 92
 Spring, 230
 Sprint.NET, 230
 SQL, 243
 SQL Injection, 254
 SQL Server Agent, 257
 sekwencja, 250
 StackOverflow, 78
 stagnacja, 94
 stała, 290
 stan, 236, 328, 329
 globalny, 334
 lokalny, 334
 zarządzanie, 334
 standard
 EcmaScript, *Patrz:* EcmaScript
 ES2015, 289
 strategia, 236, 240
 OnPush, 323
 strict mode, 277, 281
 Structured Query Language, *Patrz:* SQL
 StructureMap, 230
 struktura, 131
 strumień, 325
 obserwowalny, 296, 319, 325
 subskrypcja, 320, 321
 zdarzeń, 296
 stub, 233
 Stylus, 268
 synonim, 256
 system
 dostarczania treści, *Patrz:* CDN
 SQL Server, 243
 Transact-SQL, 243
 typów statycznych, 301, 302
 zarządzania bazą danych, *Patrz:* DBMS
 szablon T4, 174

Ś

śledzenie zdarzeń, *Patrz:* zdarzenie
 śledzenie

T

tabela, 209, 244
 indeks, 246
 kolumna, 250, 251
 rekord, 251, 252
 rutingu, 161, 182
 tworzenie, 251
 tablica, 117, 124, 276, 277
 nieregularna, 126
 providers, 316, 317
 rutingu, 169
 wielowymiarowa, 126
 wycinek, 280
 tag tbody, 266
 technika
 DDD, 229
 TDD, 159, 229
 technologia, 41, 46, 47, 65, 95, 96
 ASP.NET, 159, 176
 ORM, *Patrz:* ORM
 potencjał rynkowy, 48
 Windows Presentation Foundation,
 Patrz: technologia WPF
 WPF, 207, 209, 211
 dokowanie, 213
 rozmieszczenie elementów, 211,
 212, 213
 siatka, 211
 stos zawijany, 212
 styl, 218
 szablon, 218
 współrzędne, 213
 zasób, 214, 215
 template driven form, *Patrz:* formularz
 szablonowy
 template method, *Patrz:* metoda
 szablonowa
 test, 229
 akceptacyjny, 231
 black-box, 233
 dymny, 232
 integracyjny, 231
 jednostkowy, 159, 162, 231, 232, 233
 korytarzowy, 232
 regresyjny, 232
 systemowy, 231
 white-box, 233
 wykonywany przez klienta, 231

Test Driven Development,
Patrz: technika TDD
tracing, *Patrz:* zdarzenie śledzenie
transpiling, 340
tree of React elements, 329
tree shaking, 344
trigger, *Patrz:* wyzwalacz
truthy, 278
tryb strict, 277, 281
typ, 301
 anonimowy, 130
 any, 303, 308
 bool, 126, 134
 boolean, 275
 byte, 126, 134
 char, 126, 134
 decimal, 126, 134
 discriminated union, 311
 dopuszczający wartości zerowe, 128
 double, 126, 134
 dynamiczny, 137
 float, 126, 134
 inferencja, 302
 int, 126, 134
 intersection, 310
 literały, 309
 long, 126
 never, 314
 null, 275
 number, 275, 283
 object, 275
 ogólny, 127, 137
 Partial, 313
 prosty, 119, 126, 134
 referencyjny, 120
 sbyte, 126
 short, 126, 134
 string, 134, 275
 konwersja na typ int, 135
 symbol, 275
 Symbol, 297
 uint, 126
 ulong, 126
 undefined, 275
 union, 308
 unique identifier, 249
 ushort, 126
 void, 303
 wartościowy, *Patrz:* typ prosty

warunkowy, 313
wyliczeniowy, 128, 145
 zmiennej tablicowej, 303
type guard, 312
TypeScript, 301
 klasa, 304
 kompilacja, 301, 340
 noImplicitAny, 308
 opcje, 307
 strictNullChecks, 307
typowanie nominalne, 310

U

UAT, 231
umowa, 105
 o pracę, 49
unboxing, 147
under-posting, 192
union type, 308, 310
Unity, 230
unobtrusive, 169
URL Helper, 163
User Acceptance Testing, *Patrz:* UAT
usługa, 316
uwierzytelnianie, 174

V

value converter, *Patrz:* konwerter wartości
View Model, 234
virtual DOM, 328, 329
visitor, *Patrz:* odwiedzający

W

walidacja, 167
wartość
 domyślna, 294
 null, 274, 275
 undefined, 274, 275, 278, 281
wątek, 121, 204
 główny, 206
 pula, 154
 startowanie, 153
 zatrzymywanie, 154
wiązanie
 bezwzględne, 217
 danych, *Patrz:* data binding

- wiązanie
 - dwustronne, 318
 - względne, 217
- widok, 159, 160, 172, 233, 234, 248, 250, 316, 317
 - częściowy, 169
 - Display Template, 172
 - Edit Template, 172
 - indeksowy, 261
 - Partial View, 172
- wielopostaciowość, *Patrz:* polimorfizm
- wielowątkowość, 150, 206
- więzy
 - klucza głównego, 246
 - spójności, 246
- windowwanie, 282
- własność
 - background, 270
 - color, 270
 - CSS, 263
 - display, 265
 - float, 266
 - height, 263
 - position, 264, 267, 270
 - visibility, 265, 270
 - width, 263, 270
 - z-index, 266
 - display, 267, 270
 - float, 267
 - height, 270
 - offsetHeight, 270
 - offsetWidth, 270
- właściwość, 117
 - Attached Property, 208
 - AutoSize, 193
 - BackColor, 205
 - BackgroundImage, 205
 - CharacterCasing, 203
 - CheckState, 195
 - ContextMenuStrip, 203
 - Dependency Property, 208
 - DialogResult, 198
 - DropDownStyle, 202
 - FormBorderStyle, 205
 - HelperPage.IsAjax, 171
 - HorizontalAlignment, 218
 - Image, 197, 200
 - ImageAlign, 200
 - InnerException, 140
 - Length, 201
 - Lines, 201
 - Mask, 202
 - PasswordChar, 198
 - Rtf, 197
 - SizeMode, 196
 - SortDescription, 209
 - StackTrace, 139
 - TransparencyKey, 205
 - UpdateSourceTrigger, 220
 - VerticalContentAlignment, 218
 - WindowsDefaultBounds, 200
 - WindowsDefaultLocation, 200
- wstrzykiwanie zależności, 230, 317
- wyjątek, 139, 140
 - HttpResponseException, 184
 - obsługa, 184, 185
 - wewnętrzny, 140
- wynagrodzenie, 42, 101, 102
- wyrazenie
 - arrow function, 290, 291, 306, 331
 - EXISTS, 256
 - IN, 256
 - lambda, 149
 - WITH(NOLOCK), 257
- wywołanie
 - asynchroniczne, 146, 147
 - zwrotne, 147
- wyzwalacz, 259
- wzorzec
 - CQRS, 227
 - czynnościowy, 236
 - kreacyjny, 235
 - Model-View-Controller, *Patrz:* wzorzec MVC
 - MVC, 159, 160, 162, 184
 - kontroler, *Patrz:* kontroler
 - model, *Patrz:* model
 - widok, *Patrz:* widok
 - MVP, 233
 - model, *Patrz:* model
 - prezenter, *Patrz:* prezenter
 - widok, *Patrz:* widok
 - MVVM, 234
 - podziału odpowiedzialności, 162
 - strukturalny, 235

Z

- zakleszczenie, 255
- zakotwiczenie, 194
- zależność
 - odwrócenie, *Patrz:* odwrócenie zależności
 - wstrzykiwanie, *Patrz:* wstrzykiwanie
- zapytanie
 - do zdalnego serwera, 283
 - wydajność, 256
- zasada
 - ACID, 244
 - DRY, 226
 - jednej odpowiedzialności, 225
 - KISS, 226
 - lepsze jest wrogiem dobrego, 226
 - minimalizmu, *Patrz:* zasada KISS
 - minimalnego zaskoczenia, 226
 - mów, nie pytaj, 226
 - odwrócenia zależności, 225
 - otwarte – zamknięte, 225
 - podstawienia Liskov, 225
 - podziału odpowiedzialności, 226
 - rozdzielenia interfejsów, 225
 - SOLID, *Patrz:* SOLID
 - unidirectional data flow, 319
 - YAGNI, 226
 - złotego środka, 38
- zasób, 214, 215
- zbiór, 295
- zdarzenie, 117, 118, 146, 147, 193, 319
 - bąbelkowane, 208
 - bezpośrednie, 208
 - componentDidMount, 336
 - componentDidUpdate, 336
 - getDerivedStateFromProps, 336
 - onclick, 284
 - Paint, 199
 - pętla, *Patrz:* pętla zdarzeń
 - render, 336
 - Routed Event, 208
 - shouldComponentUpdate, 336
 - strumień, *Patrz:* strumień zdarzeń
 - śledzenie, 187
 - tunelowane, 208
 - WM_LBUTTONDOWNBLCLICK, 205
 - WM_LBUTTONDOWN, 205
- złączenie, 247
- zmiana pracy, 32
- zmienna
 - arguments, 292
 - deklaracja, 289
 - licznikowa, 273
 - prywatna, 275
 - tablicowa, 303
 - tworzenie, 293
 - typ, 301
 - zadeklarowana jako var, 274
 - zakres, 282
 - zasięg, 274, 289
- znacznik, 268
 - address, 268
 - b, 268
 - blockquote, 268
 - semantyczny, 268
- znak
 - !!, 278
 - %, 246
 - &, 199
 - &&, 279
 - @, 143
 - _, 169
 - \, 293
 - ??, 128
 - ||, 279
 - ==, 277
 - ===, 277
 - dwukropka, 135
 - gwiazdki, 317
 - unikowy, 143
 - wildcard, 246

PROGRAM PARTNERSKI

— GRUPY HELION —

- 
1. ZAREJESTRUJ SIĘ
 2. PREZENTUJ KSIĄŻKI
 3. ZBIERAJ PROWIZJĘ

Zmień swoją stronę WWW w działający bankomat!

Dowiedz się więcej i dołącz już dzisiaj!

<http://program-partnerski.helion.pl>

GRUPA
Helion



Nie daj się zagiąć – zrób dobre wrażenie na przyszłym pracodawcy!

Rozmowa kwalifikacyjna to nie jest coś, o czym myśli się z przyjemnością, ale każdy z nas prędzej czy później musi się z nią zmierzyć. Im lepiej się do niej przygotujesz i im więcej błędów uda Ci się uniknąć, tym większa szansa na ciekawą, doskonale płatną pracę. Problem w tym, że rekruterzy rzadko biorą pod uwagę specyfikę pracy programisty, oczekując, że kandydat sam się zareklamuje i zrobi to przekonująco. Jeśli chcesz zdobyć naprawdę dobrą pracę, musisz wiedzieć, na co zwrócić uwagę, by spełnić oczekiwania przyszłego pracodawcy.

Ta książka pomoże Ci się zorientować, jak rekruterzy postrzegają kandydatów do pracy, co chcą od nich usłyszeć i dlaczego wybierają tę, a nie inną osobę. Dowiesz się także, jakie obszary — związane i niezwiązane z programowaniem — interesują rekruterów i dlaczego. Zobaczysz, co warto podkreślać, na czym skupiać się podczas rozmowy, jak obrazowo opowiedzieć o swoich projektach i planach komuś, kto nie jest specjalistą w językach programowania i ma niewielkie pojęcie o tym, co potrafisz. Ponadto zdobędziesz solidną dawkę wiedzy programistycznej i poznasz przykładowe pytania rekrutacyjne. Przeczytaj, a Twoja kariera zawodowa nabierze rozpędu — będziesz pracować nad ciekawszymi projektami i czerpać głębką satysfakcję z pracy.

Dostań się do świetnej firmy!

- Po co, jak często i czy w ogóle chodzić na rozmowy rekrutacyjne
- Wybór firmy i stanowiska, przygotowanie CV
- Jak współpracować z rekruterami
- Przygotowanie do rozmowy i kluczowe 60 minut
- Łatwe, trudne, głupie — o pytaniach rekrutacyjnych
- Programowanie obiektowe, architektura aplikacji, bazy danych i SQL
- C#, ASP.NET MVC i ASP.NET Web API
- HTML/CSS, JavaScript, ES2015+ i TypeScript
- Angular i React
- Windows Forms i Windows Presentation Foundation



Helion 



helion.pl



HELION SA
ul. Kościuszki 1c
44-100 Gliwice
tel.: 32 230 98 63
helion@helion.pl

Sprawdź nasze szkolenia!



AKADEMIA IT & BUSINESS

WWW.SZKOLENIA.HELION.PL

KOD KORZYŚCI
Sięgnij po więcej! ▶



ISBN 978-83-283-2399-5



INFORMATYKA W NAJLEPSZYM WYDANIU

Cena: 59,00 zł